

A large, stylized blue letter 'A' with a white diagonal line running from the top-left corner to the middle-right, forming the left side of the 'A'.

October 1988

Vol. 2

Nº 1

Price £1.80

Archive

The Subscription Magazine for Archimedes Users

A solid blue horizontal bar spanning the width of the page.A large, stylized blue letter 'E' with a jagged, lightning-bolt-like right edge.

RISC OS 2.0 – PCW Show Report

Matrix Procedures and Functions

Sprites Old and New

Reviews of Leonardo-256, Architext, C-Guide,
Printer Buffers, Keydefine, First Fonts, Enthar-7,
Orion, Freddy's Folly and Alerion

Hypercubes – An Exploration in 3-D

ISO Pascal Extensions / Evaluating II



Happy Birthday, Archive!

I'm not quite sure when a magazine should have its first birthday – I suppose it's really when this issue is in print but anyway we had a little party for Archive at the end of September (coinciding with Ed's birthday!). We invited several people locally who had been helping us since the early days of the magazine; accountants, bankers, solicitor, office stationery suppliers, printers, the folk from the local post office, Sue's parents and Adrian Look who has done so much to help us. That was all very fine but there were an awful lot of folk I would like to have invited but couldn't because of distance problems. I know I keep on about it, but Archive would be nothing without all its generous contributors; from those who give us single hints and tips to the most prolific of authors. Thank you all, once again.

Well, they did strike!

Yes, the postal workers did go out on strike and issue 12 was held up. Sorry about that. I hope you got your copy all right. Our income did go down rather substantially, but God did provide for us and we never actually went into the red. So, many thanks to all those of you who made a special effort to get to Norwich to collect your magazine, pay your subscriptions and/or buy some bits and pieces from us to keep the cash flowing.

I was afraid too that this issue would be delayed for lack of material since virtually all the contributions come in by post. But again, through all your efforts, God has provided for us. As you can see, there is (more than) enough material and this issue should (barring accidents in printing and finishing) be going out on time.

Will they re-subscribe?

This is the question I have been asking myself just recently. Thank you to all those who have shown their confidence in us by re-subscribing. Two people, so far, have taken the trouble to say that they will not be re-subscribing; one because he has bought an IBM (shame!) and one because he objects to us acknowledging God's part in the running of the magazine. Many thanks to them for taking the trouble to let us know but we have no more intention of moving over to IBM than we have of keeping quiet about the Person who means more to us than anything else in the world!

Thanks again for all your support.

Paul & Sue

Archive

Volume 2 • Nº 1 • October 1988

Contents

Hardware & Software Available ...2	C – A Dabhand Guide28
Comment Column.....2	Printer Driver for View B3.029
PCW Show Report.....3	Printer Buffers Reviewed.....30
Help!!!.....8	Assembly Language – 4.....31
Hints & Tips9	First Fonts Review32
Matters Arising10	Sprites Old and New33
Comms Column11	Keydefine Review.....37
Contact Box14	Dragging your WIMPs.....38
Matrix Procedures & Functions ..17	Evaluating π39
What happened to Eureka!?15	Enthar-7 Review44
Hardware Corner.....19	Games Galore45
Small Ad's.....19	Hypercubes – 3D Exploration.....46
Leonardo256 Review21	MS-DOS Column52
ISO Pascal Extensions24	Fact-File53
Architext Text Editor27	

Hardware & Software Available

• **RISC Forth** – Blue-Grey Software seems to have disappeared, but RISC Forth has been taken on by Silicon Vision Ltd who are now its sole distributors. The price has unfortunately gone up to £99.95.

• **PCB layout software** – Silicon Vision Ltd are doing a P.C.B. design system for £195 inc VAT with full auto-routing, rats nesting, surface mount capability, 0.001 inch resolution, 32" x 32" max board size, on-line help, fast zoom, pan and redraw. Test and silkscreen are also included with variable line, pad, text and grid sizes, parts libraries, block move, copy, rotate, mirror and delete options. It supports Plotmate, Graphtec and all HP-GL compatible plotters. It also includes a hi-res hard copy for Epson compatible printers. It can be run from hard disc and/or Econet.

There is now also an Entry-level version of this software which has all the features of the full version except the auto-routing. The price of the entry-level version is £99.95 inclusive.

• **BASIC Compiler** – Dabs Press are close to launching their ABC (Archimedes BASIC Compiler) written by Paul Fellows – for £99 inclusive, you get the compiler, a 40 page user guide and a disc of sample programs. It will compile BASIC programs either into absolute code or as a relocatable module and it will allow you to implement star commands. The claims for speed increase are that generally it is up to ten times faster than interpreted BASIC but obviously this doesn't apply where programs are making heavy use of disc access or operating system commands since the speed of such calls will remain unchanged.

• **Spreadsheets** – If you are trying to make up your mind which spreadsheet would best suit your needs, Gordon Taylor has written a very good comparative review in *Personal Computer World*, October 1988, page 182. He looks in detail at four spreadsheets; Logistix, Matrix-3, PipeDream and SigmaSheet (all of which can be purchased through Archive.)

• **A410's** – No, don't get too excited, they are not actually "available" as such yet, but I thought you would like to know the prognosis. Top managers

inside Acorn tell me that they are going to be ready "next year" and "hopefully as early as possible in the year", but they cannot make any promises because of lead times on components and because various elements of the development and testing are not finished yet.

• **PC Emulator up-grades** – Beebug are saying that if you send your 1.09 version, they will upgrade it to 1.20 for just £3 compared with Acorn's £15, though this latter includes the up-dated manual. (*Well, if Beebug are doing it, why shouldn't we? Feel free to send your PC disc to us for upgrading at £3 a go! Ed.*) **A**

Comment Column

Archive contributors – I don't really mind what WP format you send in your articles, but please make sure that the paragraphs are double spaced otherwise, in some formats, the paragraphs disappear which makes it rather difficult to re-constitute. Also, if you want to use First Word Plus format, it is easier for me if you save them in ASCII format, i.e. switch the WPmode off before saving. Don't forget too that I would prefer to have the articles on 640k discs so that I can put them straight into Wordwise Plus on the Master and fiddle them around prior to spooling them over to the Apple.

Longer reviews, please, says C.E.Oram. He wants to know what the products will NOT do, as well as what they will do. The problem is that it is often not until you've been playing with some software for a while that you come up with the problems. What we need, therefore, is for those readers who have bought things and tried them out on real live applications, to report back to us and criticise and add to the information given in the reviews. I know it's asking for a good deal of altruism on the part of the person who has bought something and perhaps wishes he hadn't(!) but I'm sure that with the spirit that presently exists within the Archive readership, there will be those who will rise to the challenge. Thanks in anticipation! **A**

PCW Show Report

Brian Cowan, Ian Nicholls and Stuart Turgis

All three sent contributions about the PCW Show, which I have combined. To save time in editing, I have left it all in the first person singular, so it will not be clear who wrote which bit, but as long as the information is presented, I don't think it matters – I hope you, and they, agree! It was interesting to hear the different versions of what they were "told by X at the Acorn stand"! I have tried to read between the lines to work out Acorn's actual policy and have added my own impressions. Ed.

PCW should stand for, "Preview your Computer Wares" – well certainly from an Acorn angle. Many companies were promising this, that and the other, none of which was actually ready yet. Anyway, some interesting facts did appear at the show:

Arthur's Successor

For reasons best known to themselves, Acorn have decided to name Arthur's successor, RISC OS 2.0. (Presumably 'Arthur' didn't sound 'Professional' enough?) It will be packed full of all the things that Arthur 1.2 should have been and with the added bonus of at least one new applications package. You will get 0.5 Mbytes of code on four ROMs giving you a very much improved operating system and also two system discs and a new set of manuals.

RISC OS 2.0

RISC OS 2.0 supports Co-operative Multi-tasking under a WIMP environment, with full memory protection for resident programs (at long last using one of MEMC's major features!). If one application should decide to auto-destruct, all the other current applications will be completely protected and the machine will continue as if nothing had happened. Demonstrated at the Show were two versions of Zarch running in separate windows and numerous Mandelbrot windows as well.

Presumably, background tasks can run (with no WIMP output), but again this was not clear. So, not quite the pre-emptive multi-tasking of Impulse but a definite, and welcomed, step in the right direction. The desktop may still be accessed whilst programs are running, so you may enter star commands, for

example. Obviously programs must be written to support the multi-tasking, though Acorn say that existing developers shouldn't find it too difficult to convert their programs. Of course, the programs don't have to run under the multi-tasking – they can 'hog' the whole of the machine to themselves.

Desktop Features

The opportunity has been taken by Acorn to enhance the desktop significantly to make it a much more effective front-end to all the features of the Archimedes. At last, the Archimedes will have the capability of providing a consistent user interface for all applications, in the same way as the Apple Macintosh. Having learnt how the desktop works, a user will be able to move from one application to another with great ease. The movement of files from one application to another, from one directory to another, or even from one disc to another are all possible without leaving the desktop – the file's icon is just physically dragged across the screen (using the mouse pointer) from source to destination and RISC OS takes care of the rest. If the correct disc is not present, RISC OS will prompt you to insert it.

Allocation of RAM can be altered via a menu from the desktop, without needing to press <ctrl-break> to initialise the variables. RISC OS has an analogue representation on-screen of all the memory configuration settings, with horizontal bars showing the sizes of each of the areas of memory devoted to sprites, RAM disc (yes, that's implemented in RISC OS!), fonts, etc. The length of each bar can be adjusted with the mouse and the memory allocations alter immediately. As you change them, the amount of free memory alters before your eyes.

Applications (both 'normal' and 'multi-tasking') may be launched from the desktop in more than one way and, when terminated, control will pass back to the desktop. The clock now appears to be a digital version, an alarm clock has been added and the calculator has changed as well. There is an icon on the desktop which looks like a magnifying glass –

that is exactly what it is. As you move it around the screen with the mouse, anything underneath it is magnified, whether it is text or graphics.

All aspects of the desktop may be launched multiple times, so you could have 5 digital clocks, 3 calculators etc, though if you have too many, some of the tasks will slow down a lot. If you lose track of a particular application under the welter of windows that you have opened on the screen, all you need to do is to drag its icon from the directory listing onto the icon bar at the bottom of the screen. The application can then be called up at any time by clicking on its icon on the icon bar.

Also found in the desktop, are three programs – a text editor, a painting program and a drawing program – all run under the multi-tasking environment and look very good (Acorn values them at £50 each!). The text editor can have more than one window into the same text and text may be copied between windows, merely by highlighting, dragging and then releasing the text. This was also demonstrated in conjunction with the DTP package, copying from the Text Editor into the DTP.

If you, like me, were rather un-impressed by the rather garish colours of the 1.2 OS, you will be pleased to know that RISC OS makes much better use of colour. The screen is basically in shades of grey and colour is only used to enhance various areas including the icons.

Other OS features

RISC OS contains more screen modes, including mode 21 which provides a 640 x 512 display with 256 colours (using 320k of RAM!). In fact, the user can now invent almost any new mode that he or she wishes; the VIDC is directly programmable. New graphics primitives have been added including the use of Bezier curves. If you have used the font creator utility on the Welcome disc you will have seen Bezier curves in action; they enable you to draw smooth curves by defining control points whose movement alters the degree of curvature of the curve. Their inclusion in RISC OS takes the Archimedes beyond any of its competitors. The scaled plotting of sprites is another feature of RISC OS. (See Adrian Look's article about the new sprite extension module on page 35.)

ROM modules may now be copied into RAM so they can execute more quickly and a RAM filing system has been implemented. An improved version of the 6502 emulator, called 65Host, will be available, which can allow you to run some of the programs which access screen directly such as Snapper and Monsters. There is a suggestion that, amongst other things, it should implement paged sideways RAM. This will be most useful for those applications which use Service ROMs etc.

The discs will also contain updated versions of the painting and music editing programs on the current Welcome disc as well as an enhanced sprite editor program and a patience game.

The ADFS has not escaped Acorn's search for improvements; there is now an extra disc format that allows the user to access discs by name and which requires no compacting! An improved printer driver has been added. Econet users will have the ability to send and receive mail without disturbing any tasks.

Upwards Compatibility

Acorn's press release says that "the majority of existing applications should work under the new OS" and from talking to one of the Acorn representatives on their stand, it would seem that RISC OS is a complete superset of the earlier OS, so although existing programs will have to altered in order to make full use of the new desktop facilities there should be no compatibility problems with software written for Arthur 1.2. (*Hmmm!? Ed.*)

Coincidentally, the day before I went to the show I saw a demonstration of Windows 386. This is the new multi-tasking operating system for the latest generation of IBM/Intel microcomputers. Compared with RISC OS, Windows 386 went at a veritable snail's pace!

Price and Availability

RISC OS is ready now, they said, (that's not what Laurence Hardwick said at Econet '88 earlier in the same week!) but it takes 20 weeks to put it into ROM and that is the reason we have to wait until next year to obtain it.

RISC OS guidelines should be available to software developers in the near future and they will also be

getting advanced versions of the new OS as soon as it has been finalised and sent for ROMing.

The OS itself will be available to the general public in April 1989 for £29 but after a few weeks, the price will increase to £49: Acorn are very keen that we should all move to RISC OS and, having seen it in operation, I will be at the front of the queue! It will then be supplied as standard in all new Archimedes.

Desktop Publishing

Acorn's DTP package based on TimeWorks from Cambridge based GST will hopefully this will be ready by Christmas and will be the first full application to run taking advantage of RISC OS 2.0. (If RISC OS isn't available until April 89, it's not certain whether the actual release will be with the new OS, or before). I unfortunately missed the mini-seminar on the DTP, but from what I gathered the package will have:

Style sheets for individual pages. Within a page, each paragraph may also be given a type style, justification, indent etc. Many typefaces will be available, in many different sizes (but none have apparently been licensed from Adobe, so are presumably not Postscript compatible?). A set of symbols (arrows, stars etc) is also included, scalable to any size. Bullet list may be created and kerning is supported as is the adjustment of the vertical distances between lines. Left and right pages may have different layouts and page numbering is automatic. Hyphenation is automatic and reformatting, as you would expect, occurs very quickly. Pictures may be imported from either the drawing or paint package (and thus any sprite) and may be cropped and scaled to any size. A large number of printers are supported, including 24-pin dot matrix as well as laser (though there was no mention of whether the ARM based laser card for PC's would be converted for the Archimedes). No price has been fixed, but Acorn are keen to keep the price within reach of home users and thus extremely competitive with its PC counterparts.

More DTP

Whilst not being up to the standard of the full Acorn DTP package, the AVP DTP system called Pixel Perfect is available now for £79.50 with fontpacks at £24.50 and Maths, Art and Biology packs for

£29.50 each containing all sorts of useful icons for including in your published documents.

Pro-Artisan

Clares were demonstrating their new art program, Pro-Artisan. Given the price of £169.95, you would expect something significantly in advance of the best-selling Artisan, and so it is. Pro-Artisan uses mode 15 with 256 colours instead of Artisan's 16 (but you can import existing mode 12 pictures). All of the facilities of Artisan are provided together with a user-defined grid lock, a whole variety of fill options (including graduated, sprite and circular fills), a solarising function and cut and paste of irregular as well as rectangular shapes. There is a powerful distortion function and 26 in-built fonts with a font editor. Palette switching has been implemented. Screens can be loaded and saved in a compressed format, Bezier curves are supported, fills may be stepped or graduated, cut and paste will work on irregular sized objects, the distortion routine has been enhanced. Dumps are included for Integrex, HP Paintjet, Epson and HP Laser compatibles. Clares are also releasing Render Blender for £79.95 which should enable users to produce similar pictures to those found on the two Clares demo discs, complete with sound and animation of frames.

From Dabs Press

Dabs Press will be releasing ABC, a BASIC compiler, in October. Written by Paul Fellows (Ex-Head of the Arthur Development team), the compiler supports most BASIC keywords and can target the code into absolute code or a relocatable module. The f.p.u. is supported as is in-line assembly language. ASM (Archimedes System Manager) is being released in December, by the author of the Arm Assembly book, Mike Ginns. This is a toolkit of useful commands including the standard disc and memory editors but, more interestingly, OS cut and paste facilities, un-delete file capability, intelligent format and backup, screen dimmer and background printer spooling. Price for ABC is £99, ASM £49.95. Also, look out for books on Archimedes OS, BASIC V and Archimedes First steps. They are also supposed to be releasing two more pieces of software, one is a compendium of the three board games, Draughts,

Reversi and Backgammon, called "Arcendium" and priced at £14.95.

Armadillo Sound Sampler

Armadillo are producing a 16-bit sampler, which can sample at 44 kHz and oversample 4 times on playback, thus producing CD quality sound. Complete with fairly extensive software, midi in, through and out ports and Philips/Sony "CD" standard digital output and headphone socket with volume control. This is obviously aimed at more professional users – hence its price £999+VAT!! (This is only a 'special introductory price' for a limited period. If you are interested, you will need to contact them fairly quickly.) Armadillo are saying that the hardware and the first three items of software – Midi controller, Midi sequencer and Playback sequencer – will be available in November and that other software including Fourier analysis, realtime and non-realtime digital signal processing and sound synthesis will be available variously from January through to the middle of next year.

EMR's Studio24

EMR were previewing their Studio 24, to be available hopefully at the beginning of October for £99. This package can handle up to 24 tracks, though this number can apparently be multiplied up to allow even more. Unfortunately, in order to get through all his packages in the 20 or so minutes, Mike Beecher only spent a few minutes on each and so I couldn't glean too much about this package. From what I could see, and especially hear, this package looks brilliant and will certainly put the Archimedes on the 'map' for serious music!

Mike Beecher of Electromusic Research was on the Acorn stand demonstrating his company's new development in the Arpeggio suite; Studio 24 Plus. This is a MIDI recording, sequencing, composing and editing system and is a development of EMR's well-established MIDItrack system which is available on a wide range of computer hardware. It is hoped to be able to review Studio 24 Plus in the near future: Mike reckoned he had about another week's more work to put into the product before it went on sale. It will cost £99. Additional "Creations" discs for use with Soundsynth are under development and will contain sounds

captured with 8-16 bit resolution. An ARC sequencer is soon to be released to provide 8 track control of internal or external MIDI voices (£29.95).

Other new products

AMS were showing on-going stages of Finesse Paint, which should be available by Christmas but no price was mentioned. No sign of any of the rumoured Archimedes games from Superior (one of which is supposed to be Repton!), but Minerva were showing another two, Orion (a Defender/Planetoid clone) and Freddy's Folly – an intriguing variation on the shoot-em-up theme with a very clever graphics-based introduction and which involves galleons, hot-air balloons and a mobile cannon!

Argonaut are supposed to be working on Starglider II for the Archimedes, but no mention of Space Adventure or their Flight Simulator! I also couldn't find Grand Slam Entertainments, who have released Terramex and, depending on the sales, have about 4 other games in the pipeline for conversion (including the Flintstones and Red October). Colton software will soon be releasing a spellchecker for Pipedream, with a quoted speed of around 40,000 words per minute!

Computer Concepts' offerings

Computer Concepts managed to get the stand right next to Acorn and were showing Equinox, their complete office automation suite for the PC! Yep the PC! Now you can see the logic behind CC developing their own OS. They've built their own ARM card using Acorn's 4 chips, called RISCcard, then they've added a Fax, scanner, laser and PC if necessary (which is relegated to being a keyboard – best use for it!) and called it Equinox. The full system should retail for £6000 and CC only need to sell 400 to recover R and D costs. The system can send faxes, do 'photocopying' via the scanner and laser, and a host of other things, including obviously DTP, and all of course, at the same time.

However, so much time has been spent developing the OS, that only two guys have been left on the DTP (which Archimedes users are interested in). It seems unfortunately that the drawing package has had to be shelved – something happened to the programmer! But the DTP/WP package should be available in April, when RISC 2.0 is released!

"Doesn't sound like the 'September' you said back at the MU show", I said.

"The delay is because 2.0 is the first version to support printing of Acorn fonts and most of our programmers are concentrating on Impulse."

"That was supposed to be finished by September!"

"It was, but what with Equinox and everything, we've had to put back the time".

To be fair to CC, Impulse did seem a very adventurous task and I would have been really surprised if they had finished it by now, especially when you consider that they are also working on Equinox, Spellmaster for Archimedes, Z88 et al and the fact that Impulse is a virtual memory, pre-emptive multi-tasking OS, which will use a hard disc for extending memory by swapping memory in and out. (So you might not need 4 Mbytes of RAM to get the full benefits, just a 20 Mbytes hard disc!)

From what I can gather, and nothing is firm yet, there will be two versions of the DTP. At the moment, the version running under RISC OS is being developed and then converted to run under Impulse with enhancements (but no mention of them)! To further confuse matters, it seems the Impulse version won't be sold as a DTP – instead you'll buy Impulse and get the DTP thrown in free! Anyway, I was given a good demonstration of the system so far and what I saw, I liked, very much! There are still a lot of things to be done, but I feel this will definitely be another CC winner, especially if they keep the price to around the £100 mark. Interestingly, Watford Electronics had a very low profile, occupying part of CC's stand and only showing their new Aries PC's. No sign of the digitiser (which last year was on Acorn's stand!).

More Games

Magnetic Scrolls is a company that has achieved rapid fame with graphic adventure games for the Atari ST and Amiga micros. The two best known of their adventures are "The Pawn" and "The Guild of Thieves". Reviewers have raved over them. The latest release from Magnetic Scrolls is "Corruption" (£24.95) an adventure set not in some mythical forest or underground cave network, but in the City, the "shark-like world of high finance where commercial intrigue and double dealings are

commonplace", at least according to the game's publicity material. This time, the game, which was due for release on 21 September, is being released on the Archimedes as well as the Amiga and the Atari. In future, all the Magnetic Scrolls releases will be available on the Archimedes. They plan to issue Archimedes versions of "The Pawn" and "The Guild of Thieves" in about a couple of months time. These are excellent adventures and even the unique font style used on other machines has been reproduced on the Archimedes.

Presentation Graphics

Another impressive piece of software at the show was Xample Story from Eeckhorn Computers BV which was demonstrated by ECD's own André van den Berg. It is a presentation package for preparing demonstrations and lecture presentations to replace or supplement the use of slide projectors and overhead sheets – a useful system assuming you have a large video monitor or series of monitors which can be seen by all of the audience. Xample allows you to put text and diagrams on screen very simply and gives smooth scrolling in 16 colours modes. It has its own built-in "art package" as well as the facility to draw histograms and other business graphs. It has been designed to link in with Wild Visions C+ Genlock and can also be operated remotely and can be used to control slide projectors, sound sources etc.

Don't forget SID

There was only one Master on the Acorn stand – the rest were all Archimedes. The Master was being used for running SID – I wonder why they didn't use an Archimedes? (*Sarcasm is the lowest form of wit!* Ed.) SID is now available for an annual subscription of £40 (+VAT) with a line charge of 8p per minute. Add to that, of course, the telephone charges, albeit on local rates through Fastrack, and it sounds a little on the expensive side. (See Tim Saxton's article on page 11.)

Extracts from the Catalogue

The other important item that I acquired at the Show was a copy Acorn's new Archimedes hardware and software catalogue. The range of software and hardware is becoming really rather impressive. Scanning through the catalogue, I noticed that,

apart from the software of which I, as an ardent Archive reader, was aware, there is a lot more which the developers have obviously not told Paul about. *(If you know of any soft/hardware that has not been mentioned in Archive, please tell the suppliers to get in touch with us as soon as possible, for their own benefit, as well as for the other readers. We will happily send them a leaflet telling them how they can get their products known through Archive. Ed.)*

Here are a few of the items that caught my eye as I scanned through the catalogue. There are so many different suppliers for this section that I don't expect Paul will put them all in the FactFile.

Render bender – Clares Micro Supplies, £79.95 – a 3-D animation package for creating ray-traced pictures. Three **genlocks** systems are now available from Wild Vision Ltd, The Soft Option Ltd and Abbey Audio. **RGB video encoder** (£295 from Triangle TV) – takes the RGB output from the computer and generates a composite colour video signal and a TV signal. **Spelling checker** from Brainsoft Ltd (£60) – 270,000 words (cf SpellMaster with 'only' 57,000 words!) **Ophthalmology** – a whole range of software packages produced by the Institute of Ophthalmology. **Cognitive Rehabilitation Suite** (f.o.c.) from Burden Neurological Institute – helps to challenge and stimulate head injury patients. **Microplate Data Handling** package from Novo Biolabs Ltd has an interface to flow Multiskan MCC/340 and Molecular Devices VMAX photometers and offers polynomial 4-parameter logistic curve fitting! (Sorry I couldn't resist that!) Little Smalltalk from Smalltalk Express (£7.50 or £32.50 with book). Two **BASIC Compilers** Silicon Vision Ltd £99.95 and Dabs Press as mentioned earlier. (The latter is in the Books section, not the Languages and Utilities section, if you try to find it when you get your copy.) Various **precision drawing / CAD** packages, some of which have been mentioned in the magazine. Under the "specific markets" heading were programs for farmers, bakers, milkman, video rental shop owners and CNC machine tool users. The CNC software from Wyre CNC Systems Ltd at £1,000 including VAT, training and installation got the prize for the most expensive piece of software. Educationalists will find a great deal of software to whet their appetites starting from

School Administration and going through education curriculum areas from Biology to Sociology. Games included **Super Golf** £12.95 from Squirrel Software and **Cribbage** from MicroAid £6.84 inc VAT. The address list at the back of the catalogue, which includes phone numbers, is an extremely useful reference work in itself. I suggest you get a copy right away. "To get your copy of the catalogue" Acorn say, "contact your local dealer." My local dealer hadn't even heard of it, another dealers had a few in stock that he had picked up at the PC Show and the third knew that they existed and was expecting them to appear from Acorn any day. If you're desperate and cannot get to a dealer, Customer Services may be able to send you a copy, but it really is worth getting. **A**

Help!!!

- **NEC CP7 colour printer** – Has anyone got a printer driver for this printer? J.M.Smith.
- **Solidisk TTX Adaptor** – Has anyone had any success downloading software or other information using the Solidisk TTX adaptor? Denis Carey.
- **Using the RS423** – M B Wood (RAE Bedford) wants to use the RS423 interface for implementing a graphics terminal emulator. How does he send a line-break signal and how does he send an X-OFF signal when the RS423 input buffer is almost full? He's tried using OSBYTE 156 for the former and intercepting the character entering buffer event for the latter. Neither works reliably. Any ideas?
- **Colour monitor problems** – Has anyone had problems with arcing flashes on the Acorn colour monitors? i.e. a build-up of static causing a clicking noise from inside the box accompanied by white flashes on the screen. Or has anyone had dark blotches on the screen? Or areas of the screen darker than the rest? If so, let us know and tell us whether you have managed to get anything done about it, and by whom. (G.Summers, South Shields.)
- **Reading non-standard discs** – Does anyone know how you can read (and write?) 3.5" discs of other computers' formats on the Archimedes? In particular, R.M.Nimbus and Apple Macintosh.
- **Least squares polynomial** – C.E.Oram needs a machine code least squares polynomial to cope with

Hints and Tips...

• **How to wipe a disc clean** – If you want to wipe a disc completely clean, you can use `*WIPE *` `~CFR<return>` (Use this command with care – there's no going back once you've pressed `<return>!`) – in other words, remove files with any name (*), don't ask for confirmation of each wipe (~C), force them (F) to be deleted even if they are locked and do it recursively (R) so that the contents of all directories are deleted. However, even this drastic measure won't allow you to delete a library directory if one exists. So do a `*Lib ""<return>` before the `*WIPE` command.

• **GammaPlot/System Delta Plus** – If you have been trying to import SDP files into GammaPlot and have been unsuccessful, read on... Firstly you have to copy the System Delta Plus software onto your GammaPlot disc. Then if it still doesn't work, it will be because you have an older version of System Delta Plus. Send the original SDP disc back to Minerva and ask for an up-grade.

• **Mounting problems (sic)** – Do you find that sometimes it refuses to respond to a `*MOUNT` command even if you take the disc out and put it back in again? Two solutions: type `*BYE` and then you should find that `*CAT` will cause it to access the disc OR if, like me, you've got `*MOUNT` on a function key, just do a `*MOUNT` with no disc in the drive then `*MOUNT` with the new disc in place. If you are in First Word Plus where you cannot issue star commands anyway, the latter technique should do the trick since you can achieve the equivalent of `*MOUNT` by clicking on the close box (the X) in the

file menu. (Just as a matter of interest, does anyone know why it gets into this state sometimes?)

• **Users of C (and other compiled languages)** may find the following command file useful, as sent in by Peter Linstead. You can put it in the \$ directory of your hard disc and use it when starting a C programming session. Once you have assigned your current program's name to NAME using, for example, `*SET NAME MYPROG` you can edit the source file by pressing `<f1>`. Once out of TWIN, `<f2>` will compile and link the source code and `<f3>` will execute the run code. By using SETMACRO, you can change the working name with another `*SET` command without running the command file again. It assumes that both FPE and TWIN are in the \$.Library.

```
*| > $.Library .C
*FPE
*DIR $.ARM.BENCH
*SETMACRO ALIAS$AUTOT *TWIN
                                C.<NAME> |M
*SETMACRO ALIAS$AUTOC *CC <NAME>
                                - LINK |M
*SETMACRO ALIAS$AUTOR *RUN
                                P.<NAME> |M

*KEY 1 *AUTOT|M
*KEY 2 *AUTOC|M
*KEY 3 *AUTOR|M
*SET RUN$PATH
,%, $.ARM.LIBRARY., @.P.
*GOS
```

(I have just reproduced this from Peter's handwritten letter and I haven't got C to try it out on, so I hope I've transcribed it correctly. If it hadn't been for the postal strike I'd have sent it to David Wild first, so please bear with us if you find it is not completely correct. Ed.)

• **ANSI C Compiler** – Mike Sherratt writes... "On a very fundamental note... The manual is very terse. To compile the demo programs on the distribution diskette – Balls64, Sieve and Hello, heed the ReadMe file where it says 'RMLoad the FP emulator etc i.e. RMLoad fpe240'. Get to the directory level 'Bench' then do:

large numbers of calculations which take ages on the existing BASIC version. (*How about using a BASIC compiler? Ed.*)

• **Printer drivers for First Word Plus** – several folk say they've tried the Epson printer drivers with 'Epson compatible' printers such as the Star NL10 and not had much success. In particular, it seems that they get a single line followed by a formfeed as if the driver was sending 12 (form feed) instead of 10 (line feed). It sounds as if it should be an easy problem to crack, but no one seems to have done so yet – at least, they haven't told us if they have. **A**

*CC -Arthur Sieve

The spaces are significant, '-' is minus. The Sieve source file is in sub-directory 'c'. The executable file is now in sub-directory 'p'.

The program can be run from the star prompt by typing *Sieve. The 'c.' prefix is embedded (for lack of a better word) in the compiler, so don't try to create the prefix!! You are now on the way.

This information came thanks to Watford Electronics who used Acorn's dealer hot-line for me."

(Because of the postal strike, this has been inserted by the Ed rather than sending it to David for checking, so hopefully it will be OK!?)

• **Formatting from programs** – If you want to write a disc copier program, as we at Archive obviously did for copying the program discs, you have to find some way to get round the fact that *FORMAT asks for a keyboard confirmation. We did it by creating a file on the second drive that consisted of just <Y> and then called that is input from the program. Ingenious! But then we found the easy (undocumented) way of doing it:

```
*FORMAT O D Y
```

Simple if you know how!

• **Replacement 3.5" disc drives** – One reader had to replace his 3.5" drive and Acorn could only offer him the official second disc drive up-grade (which includes a new front fascia which he didn't need) so he bought an un-cased 3.5" drive from Matmos Ltd (1 Church Street, Cuckfield, W.Sussex, RH17 5JZ, 0444-414484) for £59.50 + £3 carriage + VAT. This drive was almost a direct fit and only required some judicious filing of the button slot to complete the replacement.

• **Debugging ARM code listings** – It is sometimes useful for non-ARM code specialists who want to type in ARM code program listings to have 'wayside pointers' as the code runs to help them debug the program and find out where they have made typing errors. Adding the occasional SWI & Inn (where nn is 30 to 39 for number 0 to 9) will print out these numbers as an indication of how far the program has got. *(Better still, buy the monthly program disc!!! Ed.)* **A**

Matters Arising

• **Hoverbod** – It seems that Hoverbod works on 1 Mbyte machines only. This is not clear either from the advertising blurb or even from the documentation you get when you buy it. We spent some time on this with subscribers to our Technical Help Services before we discovered that it wasn't just a case of explaining how to re-configure the machine!!! (Neither Orion nor Freddy's Folly work on a 305 either.)

• **RFS Boot files** – With the boot file as specified on page 9 last month, if you call up a *command from the RFS, it refuses to do so if the drive is empty. This is not great problem – I've been doing it like this for months, but I've just discovered by trial and error (not by understanding what I'm doing!) that if you change the order of the *SET Run\$Path command, you can avoid the problem. Try

```
*SET Run$Path RFS:$.,,ADFS:$.,%.
```

(instead of ,ADFS:\$.,RFS:\$.,%). Without the ",," in the middle, it won't go into the desktop. If you type *Desktop, it comes up with "badly nested error handler". If anyone out there understands what's going on and thinks it's of any further interest/significance, do write and let us know.

• **5.25" disc interface problems** – David Palmar commented last month (page 5) that he was told by Beebug that the Watford Mitsubishi drives don't work with the PC emulator. C.E. Oram says that the problem is that the Mitsubishi drives don't work properly with Beebug's interface – that puts a different complexion on things!

• **CCD Computer Services** have moved. Their new address is 5 Fryer's Court, Boxworth End, Swavesey, Cambridge, CB4 5RA. (0954-30279)

• **Ian Copestake Software** – Humble apologies, Ian, I keep leaving your address off the Factfile at the back of the magazine. It should be there this month, but just in case I forget it again, it's 10 Frost Drive, Wirral, L61 4XL. (051-648-6287) **A**

Comms Column – Whither Eureka!?

Tim Saxton

“How to join the IT revolution and stay solvent”

In the computer press these days one sees many articles asking why IT (information technology) and the communications revolution hasn't taken off at the rate expected. The reasons for failure usually given relate to cost and ease of access. Anybody who has experience of comms will probably go along with that.

Prestel, the national database where almost everyone can dial in for the price of a local call, has just increased its time charges in a way that hits the out-of-hours micro-user particularly hard. SID, the Acorn database is again accessible by most with a local call, but the time charge isn't trivial and the response time is rather slow, ensuring a fair-sized quarterly bill.

All the bulletin boards scattered around the country (Eureka! included) are local to some, but not to most, and by the time you have located the bits you want and downloaded them, it still costs a packet (excuse the pun) even if done in the middle of the night – and that is assuming you can fight your way in to the single PO line serving it in the first place. With a wife, three children and a dog, I have neither the time or money to take part in the IT revolution.

What can be done?

Let's go back to first principles. – We are using the phone lines to pass information from the user to the bulletin board, and vice-versa. The cost of our call is directly related to the time we hold the line, and of course all the time we are holding it, no-one else can access the board.

What is the information which passes to and fro during a BB session and how much of it is there?

A typical call

Let's look at a typical call I might make to a bulletin board. – Once I have managed to log-on, a stream of bits is sent to me from the bulletin board. How much information this stream contains depends on what I already know about what is being sent. If it is the

logon screen, then it is probably telling me nothing that I don't already know, i.e. its real information content is zero! If it is part of a menu page that is different from the last time I looked at it, there is some information there, but not a lot since, in theory, I have already received most of the menu page contents last time I logged on.

In my imaginary log-on I will now spend some time browsing through the various menus to find something (or nothing!) to download. Information here? – a few sporadic bytes from me with more or less sporadic replies from the board. Probably mostly redundant information is sent, as most of the menu items will be the same as last time.

When I have selected my menu item, the data stream might be mail for me that I haven't seen before, and the information content is higher, but it is still fairly predictable, in data terms, as it will probably be ASCII English text. When I ask for a program file, then we get to probably the highest information flow, but even this will almost certainly contain some redundant information, in information theory terms. Notice we are here really confusing two sorts of redundant information – the first sort [logon-screen] is redundant because I have seen it all before and therefore can predict exactly what it will say, and the second [ASCII mail] has redundancy because the computer will be able to predict at a bit level what some of it will be. If I send some text files, etc to the board, the same redundancy will exist.

Is this the best we can do?

I don't think it is. The present protocols have come about because bulletin boards started when people had hard copy printers, Teletype ASR33's and the like, running at 110 baud. No intelligence at all at the user's end. (In the terminal equipment, I mean!) Things have come some way since then, and files may be transferred using intelligent protocols, but I don't believe anything like enough use is made of the computer at the user's end of the system.

I would propose a bulletin board system in which each user's computer was part of the storage of the system and when you accessed the central

computer, it only sent you what you didn't already know. You would maintain on your disc all the indices and menus, and even synopses of all the public files, letters, etc. When you logged on, the board software would automatically send you an error-corrected packet of data that updated your files with the changes since your last log-on plus any new personal messages, so you had the latest state of the board available to you. This would represent quite a saving, as most of the data doesn't change anyway. Also the front page might be very pretty, etc, but to send it costs money, your money, and you probably know the phone number anyway! **Therefore principle one – You only need to be told what you don't already know.**

Now, what happens after you have got your updated menus etc representing what's on the board? You look at them, perhaps have a look at the synopsis, and decide which, if any, you want to download. If all the data for this operation exists in your computer, you don't need to be on line to browse through it. So, as soon as you have received your update, you drop the phone line. **Therefore principle two – Only pay BT while there is data being sent. Hang-up when there isn't.**

Perhaps you might want to reply to some of your messages, or download some of the new files. You prepare the requests and messages off – line and then instruct your board software to dial in. Now while your replies and messages are being sent to the board, it can be simultaneously sending you the bits and pieces you specifically requested. **Therefore principle three – If you pay for simultaneous two way communication, then use it both ways at once.**

It may be that when you first linked-in for your update, you had no files or messages to send. Therefore in the first call most of the data will be from the board to you. If you have a multi-mode modem, you would use the mode that gives the greatest speed from the board to you, e.g. V23 originate. However, if in the second link-up you had a lot of files to send and didn't want anything in return, you would select the reverse mode, e.g. V23 answer. In fact, if your modem mode can be changed under software, and will handle 300 baud and 1200/75 originate or answer, the line-time will

be least if you select 1200/75 originate mode for file transfer from the board to you, and then change over to 75/1200 answer mode (without dropping the line) to send your files. Principle three, which would suggest 300 baud with simultaneous transfer in each direction, is thus over-ridden by the 4 times faster transfer in one direction with 1200/75 baud, giving the equivalent of about 600 baud average simultaneous transfer. **Therefore principle four – Select the most appropriate connection mode for the data flow.**

Now a typical communication using these principles would be something like this:-

1) When I wanted to link in to the bulletin board, I would load and run the special user software that supports the board protocols. This would interrogate the data disc, so it knew all about the board state from the last link-in. (and even put up the front screen if you like!) The system would already know from previous log-ons the areas of interest I wanted details of, and the level of information I wanted. This level could vary from no information, to limited synopses of items, or even automatic full download of all new files on specific topics. [e.g. full download of anything to do with PC emulation, synopses of Pascal items and all public letters, but nothing else] I would prepare any messages, etc, off line and flag them for transmission. It would appear as if I was communicating with the board, but all this preparatory work was making up files in my computer. The software would then crunch these files to minimum size, using something like ARC.

2) Now I would request an update link-up to the board. My local software would select the best connection mode, (depending on the amount of data I was going to send) and dial the board. Once communication was established (no pretty screens, just a few bytes each way to see both ends agreed on who it was and what the latest state of each was) the board would search it's database for my new mail, and anything new in topics I had previously told the board I was interested in, and then send me files containing the updates (crunched, of course). While all this was happening I would send a file or files with my messages and requests. As soon as all this was done and there was no more data outstanding at either end, the line would be cleared automatically.

EXPANSION CARDS FOR THE ACORN ARCHIMEDES COMPUTER SYSTEM

IEEE488 INTERFACE a full implementation of the standard for automatic test and measurement systems

16 BIT PARALLEL I/O two 16 bit input or output ports with handshake lines for digital control applications

DUAL RS423 SERIAL INTERFACE for communicating with two additional RS423 or RS232 devices eg printers, plotters, instruments, etc

12 BIT ANALOGUE I/O in development

All the above high performance expansion cards are supplied with high level software for ease of use and a comprehensive user guide.

Take advantage of Intelligent Interfaces' expertise and purchase a complete Archimedes Computer System.

Officially appointed Acorn Scientific Dealer.



Intelligent Interfaces Ltd

43b Wood Street
Stratford-upon-Avon
Warwickshire
CV37.6JQ

Tel: 0789 415875

Telex: 312242 MIDTLX G



(Time for 2400 crunched ASCII characters in each direction at 300 baud would be about 40 seconds, i.e. one unit off peak from anywhere in the UK.

3) My local software would now update all the menus and indices on my disc, and uncrunch all my messages. I could now read all this off-line, see what was new on the board and reply, as I wanted, to my mail.

4) If there was a reply I wanted to send at once, or something new I wanted to download then, on request, the local software would re-link to the board, request the files and, while they were coming over, send any messages from me. When both were complete, the line would again be automatically cleared. The line time for all this would depend on the data. If I didn't need to reply at once, the software would save the messages for transmission automatically next time I linked-up.

Now this, I suggest, would keep the cost of using the board to a minimum. Only 1 unit off-peak to get the crunched updates and send a screen of text. Such a scheme has one other real advantage – as users are only logged on for a short time, the number that can get in per hour will be much greater and the 'number engaged' syndrome would be much reduced.

Higher baud rates than the 300 cited above give much greater data flow, off course. V23 1200/75 gives 9,600 characters in 40 seconds in the fast direction, and V22 bis gives nearly 20,000 characters in each direction!

What about the hackers?

There would be lots of curious hackers, of course, and they would need to be catered for in a polite manner that kept their line time to a minimum. Perhaps if the correct log-on protocols hadn't completed in 10 seconds send a short plain text message and drop carrier and the phone line.

Another factor affecting cost is the actual location of the board. Other things being equal, London is the best place, for two reasons. Firstly, the greatest number of people have 'local' call access to it and secondly, many long distance calls to London are at a special rate (b1), where you get 60 seconds for a unit off-peak instead of 45. (*No, Tim doesn't live in London! Ed.*)

Conclusion

Such a board would be a very different animal from Eureka! or most of the other boards around at the moment, but I believe it would fulfil my, and probably many other people's, IT requirements much more efficiently and cheaply than happens at the moment. The ability to select the level of transfer of information about new board items would let individuals limit their automatic data downloads to their special interests, with no or synoptic information about the rest. You could be sure that you weren't missing out – even if you were away for a couple of weeks – the system automatically knowing what you hadn't seen.

To implement all this requires some fairly fast and hairy software (particularly in the board computer), but then the Archimedes is a very fast and fairly hairy computer, so come on you 'comms software freaks (Hugo?). Has it been done? Can it be done?

I'm beginning to think about the data structures and program flows for such a system, but for reasons mentioned in paragraph one (W, 3C & D!) progress will be slow or even negligible. The purpose of this article is to throw some ideas out for comment, and criticism and perhaps even to enthuse someone to have a go!

That's a terrific idea, Tim. If someone wants to write the software, I'll buy a WS3000 or some-such and we'll be the guinea pigs, but I'm not moving to London – Norwich is too nice a place! Ed. A

Contact Box

- **Digitised speech** – contact A. Aasim, 69 Caerlon Road, Newport, Gwent, NP9 7BX.
- **Springboard users**, contact Andrew Walkland on (0980) 610391 ext 337.
- **Hard discs** – anyone interested in interfacing hard discs via a SCSI controller, please contact Ray Fox, 9 Piplar Ground, Bradford-on-Avon, Wilts, BA15 1XF.
- **Scottish Users** – The new Secretary of the Central Scotland Archimedes User Group is Douglas Bell, 28 Adam Crescent, Stenhousemuir, Larbert, Shropshire, FK5 4DQ. (0324-558862)

Whatever happened to Eureka! ?

Carl Wright & Roger Miah

"Is this the end of Eureka?!", I hear you ask; and so you should. Yet again, we have encountered problems with the running of the board (what else is new!), except that this time it's not the fault of people at ACMB software (we don't think).

Yes, we have had a few problems with Eureka! – At first we couldn't get the software to drop the carrier sent by the user on-line, so we set up (and I hasten to add, **dedicated**) an entire BBC B to the job of detecting when a user drops out and then 'pressing' (by using the cassette relay) the reset button on the Linnet modem. We thought all our troubles were over and that we could run the board with few (if any) problems, but guess what, we were wrong.

The Eureka! 'problem' story was just beginning. We were given a new copy of the ACMB software (we have had about 5 up-dated versions so far) and I confess, while I was copying this new version onto the hard disk I accidentally deleted all but about 10 of the users and, due to the fact that I hadn't backed the hard disk up, all users had to relog again (not an appealing thought to some of our members!).

We still thought that two problems with the board would be all that would pass our way, when the Modem started playing up and wouldn't accept any users at 1200/75 and only a few at 300/300! So we gave up and brought a new Modem (a Dataphone Demon II).

Now we should be alright because the software was originally designed to work on the Demon, albeit the original non-BT-approved version. Argh! We were wrong **again**. We then had to get ACMB to send the Demon version of the software to us. It

came after about three weeks. In the process of copying the software onto the hard disk I yet again deleted the present users, but believe it or not, I had backed them up – at least 100 or so of them! (At this point I would like to say that we **have** lost one or two of our Archive subscribers so if, when you log onto the board again, you could re-register I would be most grateful. Thanks.)

You may think that I am suggesting that we are now on-line again but I regret to inform you that we are **not**. The present technical error is that we have had to send the Demon II back to Dataphone, to be repaired because we found that it wasn't answering calls! Due to this postal strike we still haven't received the new Demon II (9th September) so we haven't got the board going yet. But I do promise (I know I promise more things than I should) that the board will work one day! (24th September and we've finally got in touch with Dataphone who say there doesn't seem to be anything wrong with the modem and they reckon it's the software!!!)

Downloads

At the moment I haven't been able to put many downloads onto the board (as I have been busy setting up my other BB), but they **WILL** be coming to you soon. I won't promise anything as I know I probably won't be able to complete it!

Eaton On-Line

If you want news on the whereabouts of Eureka! then it is possible to call our other BBs where you can get up-to-date information. The telephone number is (0603) 507216, the board runs at 1200/75 – viewdata only. You can key *EUREKA.0# for the relevant page.

Information Providers

We are still looking for information providers who would be willing to set up and maintain a section of information on Eureka!. This section can be anything that you, and at least a few of our other users, would be interested in. Remember, all donations are welcome – call for more details. **A**

- **European Archivists** – contact Jean-Philippe Demart, avenue Léon Jouret 87, B-7800, ATH (Belgium).
- **Norwegian Users** – please contact Atle Mjelde Bardholt, Batsmannsvn. 2, 3150 Tolvsrod.
- **French Users** – contact M. Trigueros Christian, 128B rue principale, 67920 Sundhouse. **A**

Now available

£105 from Archive

PipeDream

PipeDream is now available on the Acorn Archimedes. It provides comprehensive word processing, spreadsheet and database facilities integrated in a way only dreamed of before.

With other integrated packages, you have to divide your work into artificial sections, such as text, numbers and calculation, and database.

With PipeDream, you compose your work in the order you want to print it, with text and numbers all together in one document. Incorporate calculations directly into paragraphs of text and formatted paragraphs directly into spreadsheets.

PipeDream is ideal for all tasks involving words and numbers. From writing thank-you letters to encyclopaediae, invoices to cash flow forecasts, stamp-collection records to inventory management, film scripts to mail-shots.

PipeDream is 100% file and keystroke compatible with Z88 PipeDream and PipeDream on the IBM PC. It is also compatible with View Professional on the BBC microcomputer. You can create documents on any of these computers, transfer the files to any other and continue working, with no loss of information. No other software enables you to share your files with all these computers.

PipeDream for the Acorn Archimedes costs £99 + VAT.

It is not possible to detail all of PipeDream's facilities here. For full details or to order PipeDream call us on 0954 211472 or write to us at Colton Software, Broadway House, 149-151 St Neots Road, Cambridge CB3 7QJ.

PipeDream - power at your fingertips.

Matrix Procedures and Functions

Steve Drain

Although BASIC V has provided some interesting matrix functions there are a couple of conspicuous absences and some other desirable additions.

Any serious user of matrices will need to find the inverse of a square matrix; that is, the matrix you would need to multiply by to get the identity matrix, with 1's on the main diagonal and 0's elsewhere. Associated with this is the calculation of the determinant of the matrix; a single number derived from a complex series of multiplications of its elements. If the determinant is 0 the matrix is called 'singular' and it has no inverse. (This is similar to the reciprocal of 0 being undefined). The transpose, identity, edit, print and randomise procedures were written while working on the invert procedure and might also be helpful.

Determining the determinant

The determinant of a 2x2 matrix is simply $M(1,1)*M(2,2) - M(1,2)*M(2,1)$. For 3x3 and higher orders things get more complicated and the determinant is defined in terms of lower order determinants; it is recursive! Unfortunately a recursive calculation needs a vast number of steps and the function below manipulates the elements to make the calculation simpler so that it goes at a realistic speed.

Inverting a matrix

The inverse of a matrix cannot generally be calculated analytically, so a numerical method is used. This involves making a good guess, carrying out the multiplication and seeing how far out from the identity matrix you are. Then you use the errors to refine your guess and try again. This process is repeated until you are as near as you want to be. Problems are that if you are not close enough at the beginning you never get any closer and the rate at which you get more accurate results may be very slow. The procedure below uses a fairly simple algorithm for the first guess using the transposed matrix as a starting point. The corrections are based on Newton's method of iteration and use 4th powers of the errors which produces a reasonably fast closing rate. I have timed 20x20 matrix inversion to

1 part in 10^9 at about 10 seconds, which is quite respectable for a microcomputer. However, the initial guess algorithm is not very reliable at this order and errors may get bigger rather than smaller. I would not rely on it much above 18x18.

```
10 REM >matrix
20
30 REM MATRIX PROCEDURES AND FUNC-
   TIONS by Steve Drain 18 Aug 1988
40 REM
50 REM Find determinant; invert
   matrix; transpose matrix etc.
60
70 REM GLOBAL col%, row%
80
90 MODE 0
91 PROCmain
92 END
93
94 DEF PROCmain: LOCAL A(),B(),
   C(), dim%, det
100 REPEAT
110   CLS
120   CLEAR
130   INPUT"Enter number of
       dimensions: "dim%
140   dim%-=1
150   DIM A(dim%,dim%),B(dim%,dim%),
       C(dim%,dim%)
160   PROCrandomise(A())
170   A()=A()*10
180 REM PRINT"Enter new values or
       RETURN to leave the same"
190 REM PROCedit(A())
200   PRINT"The original matrix is:"
210   PROCprint(A())
220   det=FNdeterminant(A())
230   PRINT"The determinant of this
       matrix is: "det
240   IF ABS(det)>1E-3 THEN
250     PROCinvert(A(),B())
260   ELSE
270     PRINT"Matrix is singular"
280   ENDIF
290   PRINT"The inverse of the
       matrix is:"
300   PROCprint(B())
310   C()=A().B()
320   PRINT"The matrix multiplication
       of original and inverse is:"
330   PROCprint(C())
```

Matrix Procedures & Functions

```

340 PRINT "Press SPACE to finish,
      RETURN for another test"
350 UNTIL GET$=" "
360 ENDPROC
370
380 DEF PROCrandomise (RETURN matrix())
390 FOR col%=0 TO DIM(matrix(),1)
400   FOR row%=0 TO DIM(matrix(),2)
410     matrix(col%,row%)=RND(1)*
      SGN(0.5-RND(1))
420   NEXT
430 NEXT
440 ENDPROC
450
460 DEF PROCedit (RETURN matrix()):
      LOCAL temp
470 FOR col%=0 TO DIM(matrix(),1)
480   FOR row%=0 TO DIM(matrix(),2)
490     PRINT " ";col%," ";row%";":
      "matrix(col%,row%)" ";
500     INPUT temp
510     IF temp THEN matrix(col%,row%)
      =temp
520   NEXT
530 NEXT
540 ENDPROC
550
560 DEF PROCtranspose (matrix(),RETURN
      transpose())
570 FOR col%=0 TO DIM(matrix(),2)
580   FOR row%=0 TO DIM(matrix(),1)
590     transpose(col%,row%)=matrix
      (row%,col%)
600   NEXT
610 NEXT
620 ENDPROC
630
640 DEF PROCidentity (RETURN matrix())
650 FOR col%= 0 TO DIM(matrix(),1)
655   FOR row%= 0 TO DIM(matrix(),2)
660     matrix(col%,row%)=- (col%=row%)
670   NEXT
675 NEXT
680 ENDPROC
690
700 DEF PROCprint (matrix());LOCAL num%
710 num%=0%
720 @%=&10208
730 FOR col%=0 TO DIM(matrix(),1)
740   FOR row%=0 TO DIM(matrix(),2)
750     PRINT, (matrix(col%,row%));
760   NEXT
770 PRINT
780 NEXT
790 @%=num%
800 ENDPROC

810
820 DEF FNdeterminant (Z()) : LOCAL
      dim%,lev%,col%,row%,factor,total
830 dim%=DIM(Z(),1)
840 total=1
850 FOR lev%=0 TO dim%-1
860   FOR col%=lev%+1 TO dim%
870     factor=Z(col%,lev%)/Z(lev%,
      lev%)
880   FOR row%=lev%+1 TO dim%
890     Z(col%,row%)-=factor*Z
      (lev%,row%)
900   NEXT
910 NEXT
920 total=total*Z(lev%,lev%)
930 NEXT
940 =total*Z(dim%,dim%)
950 END
960
970 DEF PROCinvert (matrix(),RETURN
      inverse())
980 LOCAL errors(),identity(),dim%
990 dim%=DIM(matrix(),1)
1000 DIM errors(dim%,dim%),identity
      (dim%,dim%)
1010 PROCidentity(identity())
1020 PROCtranspose (matrix I, inverse())
1030 inverse()=inverse()/FNfactor
1040 REPEAT
1050   errors()=inverse().matrix()
1060   errors()=identity()-errors()
1070 IF FNfinished THEN ENDPROC
1080 PROCmake_corrections
1090   errors()=identity()+errors()
1100   inverse()=errors().inverse()
1110 UNTIL FALSE
1120 ENDPROC
1130
1140 DEF PROCmake_corrections: LOCAL
      cor
1150 REM local to PROCinvert
1160 FOR col%=0 TO dim%
1170   cor=errors(col%,col%)
1180   cor=1+(1+(1+(1+cor)*cor)*cor)
      *cor
1190   FOR row%=0 TO dim%
1200     errors(col%,row%)=errors(col%
      ,row%)*cor
1210   NEXT
1220 NEXT
1230 ENDPROC
1240
1250 DEF FNfactor: LOCAL col_tot,
      row_tot,col_max,row_max
1260 REM local to PROCinvert
1270 FOR col%=0 TO dim%

```


Hardware Corner

Brian Cowan

It is a somewhat shorter column this month as I still have not quite recovered from my holidays! I am grateful to Mike Harrison who has provided much useful information.

A300 and A400 series p.c.b.'s

We must lay to rest one misconception once and for all. The boards in the 300 machines are not the same as in the 440's, though they do use the same ROMs.

RAM Capacity

First let's look at the question of RAM. This follows on from my last Hardware Corner in the August issue. The 305 and the 310 use 256k by four bit chips whereas the 440 uses 1M by one bit RAMs. Clearly the data bus wiring is vastly different in these two cases. A further consequence of this is that while the 305 and the 310 use the same board, the model 410 (if/when it appears) must use a completely different board. Thus we can understand the delay in the introduction of this model.

So what does this mean for memory upgrades beyond the official one meg of the 300 family? It is a relatively (?) simple job to make a header board to interpose between the MEMC chip, the VIDC chip and the circuit board. In this way the address lines and the data lines can be intercepted and diverted to RAM on this board. There are, however, two disadvantages to this scheme. Firstly the cost of four megs of RAM is prohibitive at the present, and secondly the machine's own one meg or half meg of RAM is wasted. A scheme which used this memory as well would be difficult to implement and its functioning would depend on the operating system

used since the machine would have to recognise a non-standard memory configuration.

Video Differences

Another difference between the 300 machines and the 400's is in the area of (ultra) high resolution monochrome graphics. It is a publicised fact that there is hardware facility for high-res mono on the 440, although there is no software support for this. Apparently there is also high-res mono circuitry on the 300 machines. This is different from that on the 440, and so far as I know, undocumented.

All in all, the 300 and the 440 boards are quite different. This leads to the question: what are all the vacant spaces on the 300 series boards for? This concerns mainly the hard disc controller. It is my guess that fully populated, these boards were used inside Acorn as early development models. It may well be that some time in the future a new board is introduced for the 305 and 310 with no spare spaces.

Maybe some enterprising reader will investigate installing a Winchester controller directly on the 300 board!!! **A**

Small Ad's

Logistix – almost brand new – only £75 – contact Norwich Computer Services on 0603- 507057.

Programs transferred confidentially – £1.50 (or £4 including disc). DFS, DDFS (Watford) – ADFS 5.25" or 3.5", or have your BBC BASIC programs transferred to IBM 3.5" or 5.25" or from 3.5" to 5.25". A. Asim, 69 Caerlon Road, Newport, Gwent, NP9 7BX.

```
1280 col_tot:=0: row_tot:=0
1290 FOR row%=0 TO dim%
1300   col_tot+=ABS(matrix(col%,
                                row%))
1310   row_tot+=ABS(matrix(row%,
                                col%))
1320 NEXT
1330 IF col_tot>col_max THEN
      col_max=col_tot
1340 IF row_tot>row_max THEN
      row_max=row_tot
1350 NEXT

1360 =col_max*row_max
1370
1380 DEF FNfinished
1390 REM local to PROCinvert
1400 FOR col%=0 TO dim%
1410   FOR row%=0 TO dim%
1420     IF (ABS(errors(col%,row%))
              >1E-7) THEN =FALSE
1430   NEXT
1440 NEXT
1450 =TRUE A
```

PAINT YOURSELF A PROGRAM

If you're a programmer looking for an easy way to incorporate sophisticated graphics into your programs, ArcImEd is the perfect answer.

ArcImEd is a comprehensive graphics editor, written specially for the Archimedes, which allows you to draw pictures and store them. Not as pictures – but as programs!

Using the mouse, pop-up menus and the function keys, you can build up your picture from solid lines, dotted lines, circles, ellipses, triangles, rectangles, parallelograms...

A user-definable grid (a sort of on-screen graph paper) helps you square up your drawing and add text.

A 'snap' (join to the closest existing point) facility makes sure that your figures are closed and your fills don't flood out.

Colours are selected from and mixed on a colour menu.

And there are built-in transformations – reflection, rotation, translation and scaling. All of them can be applied to the full screen, or to a user-defined window.

And because while you're drawing you're also writing a program, you can edit your picture, stepping backwards and forwards through the drawing, modifying whatever you want to change. Both versions remain onscreen until you ask the computer to redraw the picture, so you can see exactly what you're doing.

Once your picture's complete, you can either save it to disk or to memory as a BASIC program. And to ensure that the routine will fit straight into an existing program, you can specify – as you save – whether you want to retain existing colour statements, whether you want to save co-ordinates as absolute or relative, and whether you want the picture saved as a procedure with built-in parameters.

The whole package is fully documented in a unique online manual that you can call up onscreen at any time. You can either step through it, page by page, or call up the index and use the mouse to select the topic you want: the correct page will appear onscreen.

ArcImEd costs £24.95, including VAT, postage and packing.

GIVE YOUR ARC A HAND

Servo, specially designed for the Archimedes, allows you to control up to four servos (the type used in radio control cars and aircraft) from your computer.

In effect, it gives your Archimedes up to four 'hands' that it can use – with a little mechanical assistance – to control outside objects like switches, locks, doors, windows...

Servo consists of a small hardware unit which fits on the Archimedes' printer port, a software module and an 'online manual'.

You supply the servos (and the batteries which your servo needs).

The online manual explains how servos work, shows you how to wire them up, and provides an interactive test environment – so you can check that everything's working properly and you can experiment with test values to see what suits your particular servos.

Servo costs £29.95 including VAT, postage and packing.

Please send me:

☐ copy/copies of
ArcImEd at £24.95
(ArcImEd will be
despatched by return
post).

☐ copy/copies of
Servo at £29.95 (Please
allow 28 days for
delivery).

I enclose a cheque for £
made payable to Jansons.

Name

Address

Postcode Tel

Jansons

93 North Street,
Stilton,
PETERBOROUGH
PE7 3RR

Telephone: (0733)-244702

Leonardo256 – Mode 15 Art Package

Steve Bruntlett

Leonardo256 is a mode 15 art package for Archimedes computers which have at least 1 mbyte of memory. It is basically a powerful pixel editor, but it has a number of powerful drawing functions available both on the original image and on the magnified area of the image. All the drawing options can be used at any of the levels of magnification (1, 2, 4, 8 and 16).

The program starts up with an on screen menu which is used to select the drawing options and the picture and font filing options. The menu disappears automatically when the pointer is moved off the menu.

Draw is used by pressing <select> to plot individual points and for freehand drawing. Pressing <adjust> whilst in this option will pick up the colour underneath the pointer and enable you to start drawing with that colour.

The 256 available shades of colour are displayed in the central area of the menu box. The current colour is displayed in the long box at the top of the palette. The position of the current shade of colour within the palette is also indicated by being high-lighted with a black border.

Fill works like most other fill routines, but pressing <adjust> will pick up the colour under the pointer and enable you to fill in that colour straight away without having to go back to the menu and figure out exactly which of the 256 different shades of colour you were working with.

Line can be used to draw single or continuous lines.

Rectangle, Triangle, Circle, Oval & Ellipse may be drawn in outline or solid in any of the 256 shades of colour. The ellipse option allows you to put the ellipse at any angle, whereas ovals are always aligned with their axes vertical and horizontal.

Text allows you to enter text from the keyboard and then position and fix it on the screen. There are eight different fonts available in a variety of styles and point sizes. These look to be very useful for poster type work.

Airbrush allows you to spray points of colour in a random, centre-weighted pattern. <adjust> can be used to change the size of spray area.

Copy, Flip X & Flip Y are used to copy parts of the screen to new screen positions either in the same orientation or flipped vertically or horizontally.

Magnification

<Select> allows you to select a magnified area to work on with a moveable rectangular outline indicating the current magnified area. The size of the rectangle can be adjusted with <adjust> without having to go back to the menu.

Remember that all the drawing functions are available at any degree of magnification, which is what sets Leonardo-256 apart from most other drawing packages.

Print is used to dump the screen to an Epson FX compatible printer.

If you want to dump screens in colour, you will need to buy a separate colour dump such as the ones from Musbury Consultants. They do dumps for both the Integrex 132 and the Canon PJ1080A colour ink jet printers. (*Available through Archive. Ed.*)

Undo will remove the effect of the previous drawing operation. All the drawing operations since the last use of the menu will be undone, so if you don't want to keep going back to say the start of a complicated fill operation, bring up the menu from time to time to limit the number of operations the Undo command will remove.

Undo can always be reselected to bring back the original screen, i.e. to undo the undoing you've that just done!

Undo is not affected by <select>, *, Qsave, Save, 16x, 8x, 4x, 2x, 1x and Print, so you could, for example, draw a shape using a high magnification, then look at it in low magnification before deciding whether or not to use Undo to erase it.

The * option permits operating system commands to be entered.

Fload is used to load in a font file from disc.

Qload and **Qsave** are used to load and save pictures in compacted form. Since you can only get 4 un-compacted screens on an 800k disc, this facility is extremely useful and much quicker than the normal loading and saving method.

Load and **Save** allow you to load and save screens in the normal way.

x16, x8, x4, x2, x1 are quick ways of changing the magnification of the image, as an alternative to using the <select> option.

Font Generation allows you to create new fonts or amend the fonts supplied with Leonardo-256. Again this is covered in great detail in the manual.

Using Leonardo-256 with the Watford Video Digitiser

Leonardo-256 can be used easily with the Watford Video Digitiser by using operating system commands, using the "=" prefix to direct the output to the image screen. The best way to access these

commands is to set up the function keys. This is covered in detail in the very comprehensive and well presented manual.

Conclusion

Leonardo-256 is very easy to use. There is very little to go wrong. I experienced no problems in using the program. While this program hasn't as many facilities as Artisan, it has the advantage of the extra 240 shades of colour that mode 15 offers compared with mode 12.

Being able to work and use all the drawing facilities on the magnified sections of the screen is the main advantage of Leonardo-256. If you need a graphics program to use with the Watford Video Digitiser, then this fits the bill. It could be used to tidy up digitised screens, add text to them and use the resulting screens for presentation purposes.

The program is available from Beard Technology at a cost of £19.50 inclusive. **A**

Leonardo

Archimedes Mode 12 Art Package

A comprehensive set of drawing options all available at four levels of magnification

Quick compressed screen files, so you can save many more screens on a disk

Available for all models of Archimedes

£17.50 inclusive

Leonardo 256

Archimedes Mode 15 Art Package

Mode 15 version of Leonardo with 256 colours plus a font generator

(Leonardo 256 requires 1 megabyte of memory)

£19.50 inclusive

Available from: **Beard Technology**
111 Evering Road
London, N16 7SL.

SILICON VISION

ARCHIMEDES SOFTWARE

SolidCAD

£49.95

New

The ultimate 3D Draughting System for Architectural design, Interior design, Engineering Design and Teaching CDT. Allows drawing in plan, front & side elevations and also directly in 3D view. Includes powerful zoom & pan options for precision draughting and surface definition for creating Solid colour objects. Also includes Sweep, Extrude & Macro facilities for designing very complex objects easily. Designs created with SolidCAD are compatible with the Realtime Graphics Language for high-speed flicker-free animation. Also performs smooth shading for realism. SolidCAD users can upgrade to the Realtime Solids Modeller for £40.00.

REALTIME SOLIDS MODELLER

£89.95

New

The package includes both the sophisticated design environment of SolidCAD and the high speed animation capability of a Realtime Graphics Language module developed in pure ARM risc code for supercharged performance. The RGL module can be used to create standalone flicker-free animation of designs from your own programs. Through our in-house expertise in 3D Design and High-speed techniques, no other package can rival the design environment & animation speed of the Realtime Solids Modeller.

ARC-PCB

£195.00

New

The ultimate PCB design system developed specifically for the Archimedes with a specification that cannot be matched. Includes Automatic routing, Rats-nesting, 8 layers, Surface mount capability, 0.001" resolution, 32"x32" maximum board size, On-line Help, Fast Zoom/Pan/Redraw, Text & Silkscreen facility, Variable Line/Pad/Text/Grid sizes, Part Libraries, Block Move/Copy/Rotate/Mirror/Erase options, and up to 300,000 components. Plotter and printer drivers included. An entry-level ARC-PCB system (without auto-routing) is available for £99.95 (Auto-route upgrade for £100). Enquire about our turnkey PCB design systems complete with Colour Archimedes and/or plotters with prices from £1220 to £3909 (Inc VAT).

RiscBASIC

£99.95

New

Supercharge your Archimedes Basic programs by compiling them automatically into pure ARM risc code with the RiscBASIC compiler. Features include Relocatable modules, Cross reference of all variables, functions, and procedures, Floating point and Integer support, Standalone code generator, Optimising compiler & Full runtime error handler.

RiscFORTH

£99.95

New

A new 32-bit implementation of the FORTH-83 standard, designed to take full advantage of the ARM architecture. Features include Multi-tasking, Optimising compiler, ARM assembler with floating point mnemonics, Full screen Editor, File system interface, OS calls support, Floating point & Integer maths, WIMP support, Single-step debugger, Shadow screen for documentation, Block manipulation, Dictionary/Vocabulary display and a standalone code generator.

SUPER-DUMP

£24.95

New

The ultimate printer driver which takes advantage of the highest resolution capability of ordinary Epson compatible printers to provide 1920x1024 resolution. Images can also be scaled, positioned and previewed before printing. Fully compatible with all our CAD systems. Your own graphics programs or other CAD packages can be made compatible with Super-Dump by the addition of a few simple commands. An example program is included in the package.

Gate-Array Design & Teaching System

£89.95

New

The system provides an introductory course for Silicon chip design with facilities for Array Design, Cell Design, On-line Help and a tutorial on integrated circuit design. The package is ideal for teaching purposes and provides a low cost entry system for learning chip design skills. Compatible with Super-Dump for hardcopy. Plotter drivers are optional at £19.95.

All software are custom written for the ARCHIMEDES.
All prices include VAT and Carriage (Overseas orders add £4).

SILICON VISION LTD, SIGNAL HOUSE, LYON ROAD, HARROW, MIDDLESEX, HA1 2AG.
TEL: 01-422 2274 or 01-861 2173. FAX: 01-427 5169. TELEX: 918266 SIGNAL G
(Access/Mastercard/Eurocard accepted)

Languages – ISO Pascal Extensions

David Wild

The extensions provided by Acorn are some of the most exciting features of the language and also the worst documented, so it is worth spending a little time finding out what they will do for you. Don't forget that a program using any of these extensions will need to be compiled using the '-ex' switch.

There is, of course, always a problem with extensions; they are the least portable part of a program and may lead to considerable re-writing if another computer is to be used. Fortunately the most significant extension of all, separately compiled modules, mainly changes the way in which standard pascal statements are "packaged" and so provides much improved flexibility without needing major rewriting if the program is to be transferred.

For instance, I have written a module which incorporates a number of date and time calculations and it could, if necessary, be made into an "include" file on another system. The only machine-specific part of the module is the method of getting the date from the system clock – and this would need to be changed anyway.

Probably the best way of writing about the extensions is to deal with all the minor ones first and then deal with the modules, so I will use the order of the items in the manual, except for modules. Each section will be followed by my suggestions about the best way of maintaining as much compatibility as possible with other versions of pascal.

Characters in identifiers

In order to increase the readability of programs you can insert '\$' or '_' characters in identifiers. The characters are significant and identifiers containing them are different from identifiers using the same letters but without the extra characters.

My suggestions for standards to help with compatibility are:-

- 1) Only use the '\$' in operating system related names. These names will have to be edited out anyway.
- 2) Do not use the same identifier both with and without the underscore characters. You can then edit them out if necessary without causing confusion.

Non-decimal constants

Numbers used in the program (the manual refers to them as 'constants', but this includes numbers used in expressions) can be expressed to any base between 2 and 36 by separating the base from the number by an underscore:-

2_1010 or 16_ffff.

If you are really determined, you can go beyond base 16 but you will have to use more of the alphabet to express the numbers. The potential for confusing yourself seems to grow exponentially!

- 1) Confine such numbers to constants in the standard pascal sense and use the names in the actual program. Provide plenty of comments so that the converter knows what you are trying to do.
- 2) Don't use bases other than 2, 8 or 16. Most programmers understand these and can do conversions on their calculators if necessary. It is difficult to think of a good reason for using any other – but, if you must, put the reasons in your comments.

Non-printing characters

It is possible to specify non-printing characters in character or string constants by writing the hex value of the character preceded by a backslash. Examples given in the manual include:-

```
bel = '\07';  
help = 'HELP\07\07';
```

Acorn themselves recognise that this can cause difficulties if a backslash should happen to have been used elsewhere in the program and insist on a separate switch being used at compilation time to allow it to work. As Twin will allow you to insert control characters in your source code it would seem to be unnecessary.

Compatibility suggestion: DON'T!

Bit-vector operators

This section, whose name I haven't met before, is really concerned with what I have always called 'logical operators'. They can be very useful in many ways and I feel that it is a great shame that they have not yet been included in standard pascal. (By the way, has anyone managed to do a logical AND on two integers using standard pascal arithmetic operators?) There is not much to be said about these

operators, so long as you remember that they operate on whole (4 byte) integers only. They will certainly make such things as hashing routines much simpler.

- 1) *Plenty of comments.*
- 2) *Don't use them if there is a standard way of achieving the same result, unless the routine is really time-critical. I notice that Smith & Wiggins use them to build up parameters for some of the procedures in their Graphics Extension modules. As the graphics method is already non-standard, compatibility is already ruled out.*

Miscellaneous routines

There are four routines in this group:-

Linenumber – This routine returns the line number of the line on which it is written so I presume that you can put 'writeln(linenumber)' but I really can't see why you should ever want to do it, except if your program is so difficult to debug that it needs to tell you where it is.

Address – This returns the machine address of the variable supplied as a parameter, and can be very useful for interfacing with machine code extensions (see later) or operating system routines.

Size – This returns the number of bytes occupied by a variable and is especially useful with records. As with address, it is mainly for interfacing with operating system routines. I have managed to write routines for random access to files of records and the size function was invaluable.

Monitor – This causes a dump like a post-mortem dump, but lets the program carry on. As with the line number function, it is really for use when you get into deep trouble, and I am sure that Archive readers design their programs too well for that!

- 1) *Not really a problem. Linenumber and monitor should not appear in a program which is ready for distribution and the other two are machine dependent. Use plenty of comments.*

Otherwise in case statements

This "non-standard" extension seems to be cropping up in more and more implementations and I would expect it to be inside the standard before very long. The only likely problem is the different syntax that might be used.

- 1) *Include a comment to show its use.*

String assignment

This is another extension that is creeping into more and more versions of pascal and also may have slightly different syntaxes. It is very tempting since it saves the specially written transfer routines that might otherwise be needed. The specification in the manual says that 'Assignments of the form S1 := S2; where the length of S2 is less than the length of S1 are permitted', but it doesn't say what happens if you try it the other way round. You could get caught with this in parameters but the compiler should keep you out of trouble.

- 1) *Put suitable comments where you have used the facility.*
- 2) *Don't forget that conformant array parameters are part of the pascal standard, and these achieve many of the same results.*

Source file inclusion

This facility appears in a number of versions of pascal, but is normally confined to one level. In Turbo Pascal, for instance, it is specifically stated that include files **must not** contain further include statements. There must be a restriction somewhere on the Archimedes but it allows at least five levels and this ought to be enough for anybody! The biggest problem is that it is not easy to follow line numbers in the listing when you use more than one level of inclusion.

- 1) *Don't go mad, but you can always use a text editor to follow the inclusions through, and so make up a flat file. Providing you are not pushed for disk space this should be no more than a tedious chore.*

Assembly language statements

Although the manual says that the use of assembly language should be only as a last resort it is, at present, the only way of communicating with the operating system. The principal point that is not given in the manual is that all the assembly language statements must be in upper case. If you forget this you will have some interesting looking error messages! There is also a bug in the compiler which means that you must put a semi-colon after every assembly language statement even if the syntax does not require it expressly.

- 1) *As assembly language is machine dependent compatibility is not a major problem. It saves a lot of hassle, though, if you confine assembly language to procedures and functions that are*

obviously Archimedes, even if it means an "unnecessary" call.

File binding

One of the big problems of pascal programming has been the fact that you could not test whether or not a file had been picked up correctly, and so you could have a program crash because the user had forgotten to load the correct disk. The new versions are very similar to those used in Turbo Pascal and seem to be very sensible.

An interesting extension, not explained clearly in the manual, is that you can connect a 'text' variable to the printer or to the vdu by writing 'rewrite(printfile,printer:)' or 'rewrite(screenfile,vdu:)', assuming that printfile and screenfile have been declared as text variables. If you use this method you must not declare printfile or screenfile in the program heading. If you want the files declared in the program heading, and it is good documentation, you must attach the files at the time the program is called by something like 'myprog -printfile printer: -screenfile vdu:', and then omitting the file specification from the reset or rewrite statements.

If you declare the files in the program heading and then try to use the reset statement with a variable name, rather than a literal, you will get a peculiar error message saying that 'a string constant is required here' and referring you to the definitions of \$include and alias - neither of which is relevant to the problem.

The provision of extended forms of reset and rewrite with status variables is useful to allow checking, and then the printing of messages to the user asking for the input disk to be checked or changed. Don't forget that you will need to declare the status variables in the normal 'var' list.

- 1) *Put all file opening and closing statements into separate procedures. Any procedure call overhead will be negligible and will be paid for by the increased amendability many times over. Don't forget, plenty of comments.*

'Univ' parameter

Using the word 'univ' in front of the type definition of a var parameter will disable type checking. I presume that it has its uses, but the potential for danger is immense.

Compatibility suggestion: DON'T!

Order of declarations

The standard ordering of const, type, var, procedure and function declarations has been relaxed. This means that there can be more than one of any entry, although any identifier can only use those features which have already been defined. Presumably this has been done so that blocks of declarations can be put in one \$include file without generating an error.

- 1) *Plenty of documentation.*
- 2) *Use only in \$include files.*

Constant expressions

In declaring constants it is possible to use arithmetic combinations of constants already declared. This means that we can have:-

```
const
one = 1;
two = 2;
three = one + two;
```

This will work for addition, subtraction and multiplication but not for division, so its use is rather limited. The compiler will also disallow:-

```
aa = chr(65);
```

and

```
aa = chr(sixtyfive);.
```

Because of this I cannot see any good reason to use it, so there is no need for compatibility suggestions.

Short Reals

The standard real variable gives 64-bit quantities and the extensions offer an extra type, shortreal, which provides 32-bit quantities. You can compare the two types as the compiler arranges any necessary conversion. I have not found any great reason for using shortreal, unless you are very short of space - and there is no claim that the program will run any more quickly as a result of using them. Perhaps the only reason that comes to mind is testing a program to see if it will still run on a computer that offers only the shorter length.

- 1) *Plenty of documentation.*
- 2) *Use only one or the other, then the program can be changed with one touch of a text editor.*

This will be continued next month when David will turn his attention to the very important topic of modules and static variables. A

Architext – Text Editor

Michael Dunbar

Architext is a fully-featured, mouse driven text editor for the Archimedes. Written in "C", it aims to provide the optimum environment for writing source files for C, FORTRAN, assembler or even BASIC. Written by a professional programmer frustrated by the shortcomings of Twin, does it measure up?

Getting going

Starting up is simple enough – make sure you have Architext on your current disc, then type:

```
*at<return>
```

Optionally, you can include up to 5 file names, which Architext will automatically load into its separate work areas (see below). This dumps you in the main editing screen. This takes up 28 lines, giving a good uncluttered display. The bottom 4 lines comprise a number of status indicators, plus some "buttons" – more of that later. The text-entry cursor is a small red bar, moved via the cursor keys, and the editing is much as one would expect – much as any wordprocessor, except of course it will not prevent words splitting at the ends of lines.

The facilities

Architext has some surprises up its sleeve. There is another cursor, coloured blue, controlled by the mouse. Movement to any point on the screen is simple and quick. Roll the blue cursor to where you want to be, then click select, and there you are! Although this takes a bit of getting used to, it is so quick and natural that time spent just tapping the cursor keys is dramatically reduced.

Furthermore, the function keys are, wherever possible, identical to the BASIC editor, load, save, exit, undo etc all fall naturally to hand. A function key strip is included.

The status lines are another aspect of "at" that has been excellently thought out with, I suspect, a little help from mainframe editors. From left to right, the features are:

4 scroll buttons. Move the mouse cursor off the editing screen itself and it changes to the more

familiar pointer. Then click on the up or down buttons. Hey presto! The editing screen scrolls smoothly, the red cursor remains where you left it – excellent. The other two buttons perform the same thing, only a screenful at a time. The buttons are a bit small, being 8 by 24 pixels, but making them any bigger with the current screen arrangement would be difficult.

Next comes a box with a sausage! The box is an elongated rectangle, with a long white bar inside. The box, end to end, represents the whole document, and the white bar, the region of the document on screen. Pointing to halfway down the box moves you to the middle of the document, in a similar way to the scroll bars in the WIMP system. There is also a vertical line showing where the cursor is on the screen.

Finally, after a line number indicator (very useful), there is an area showing the filenames of any files in memory. There may be up to five, in separate work areas. Switching between them is achieved through the numeric keypad. The bottom line of the display is reserved for prompts and error messages.

The mouse is also used for marking text. Move to the beginning of the block to be marked, press <select> and then, holding down <adjust>, move to the end. Pressing <menu> then brings up a menu offering delete, move and copy, all of which are easily achieved. The extent of the marked block is shown in the scroll bar rectangle area.

Mentioning other features would seem like icing on the cake, but there are plenty of them: Limited undo (just like the BASIC editor), different work areas on screen at the same time, alteration of line terminator, search and replace (with pattern matching and including non-printable characters), OS commands, mode 19 support and use of the user definable keys. If a file has been loaded, it offers its name as a default when saving. It also reminds you if you have altered text when quitting. Also, and this may be the clinching factor for many people, it is fully TWIN compatible, so editing BASIC programs is a doddle. Last but not least <print> (the key) prints, sending the contents of the current work

area to the printer. This, combined with one of the currently available Archimedes printer buffers, significantly improves performance.

Documentation

The manual is one of the best I have seen, it is compact (20 pages), while leaving nothing out. The clear typeset text includes a tutorial which provides an excellent demonstration of the power of the system. Error messages are well explained and there is even an index.

Gripes?

Hardly any at all. The biggest for me concern the cursor movement. The mouse is OK, but from the keyboard, there is no quick jump to the beginning or end of a line, or word, similarly, no page up, or down. Irritating, is the fact that the cursor keys do not move from the end of one line to the beginning of the next, and vice-versa. Delete word isn't there, but there is a delete to end of line. The ability to swap case would have been nice, as would a delete to the right and the ability to mark text without the mouse. I like my cursor to be a box rather than a narrow bar and I do prefer vertical scroll bars à la Microsoft Word, but that's just me.

Finally, of course, it is not a memory-resident module. This is hardly the author's fault, but if the Acorn C compiler ever does support relocatable code, I am sure that an ArchiText module will not be far behind.

Conclusion

These are all minor niggles though and hardly detract from what is an all-round professional product that will significantly reduce the time taken to edit source files. If a user had a valid plea, I am sure that the author would be more than willing to make productive modifications. HopeSoft are even offering a demo disc for £3.50, refundable if (when?) you decide to buy the full product.

In conclusion, I only hope that ArchiText gets the publicity, and thus the popularity it deserves. As to would I buy it, the answer must be a resounding yes!

ArchiText – from HopeSoft, £24.90 **A**

C – Dabhand Guide

Stuart Turgis

When deciding on whom to give a book to review, an editor has two choices: 1) He can give it to someone who knows about the subject concerned and can thus judge the book based on their own knowledge. 2) He can give it to a complete novice on the subject and see if he/she can learn from the book without any help.

Well, just to be awkward, I fall somewhere in between the two! I am a complete novice at C but do have a sound knowledge of programming languages in a commercial environment (Pascal, COBOL, BASIC & Assembler to name but a few), so do bear this in mind.

The book is certainly not small, some 510 pages long including the index, divided into 37 chapters! The author, Mark Burgess, has obviously decided on shorter chapters and more of them, and I feel this makes reading easier – you can take breaks quite easily without having to stop mid-chapter.

Further, each of the chapters is 'encapsulated' to a high degree; they don't need to be read in strict sequence. Information about the semantics and syntax of the language is introduced gradually, some of the more complex aspects being dealt with briefly in the early chapters and then expanded upon in later chapters. Again, I found this style much more acceptable and conducive to the learning process.

Where possible, diagrams are used to convey points which are difficult to summarise succinctly and each chapter finishes with a half-dozen or so questions. (The answers are given in the back of the book.)

Some nice touches to the book are the chapters on the individual micros, highlighting the compilers available on most popular computers (including the Archimedes) and their differences; hints on errors and debugging your programs; and four full example programs (in addition to the listings found within each chapter).

Although I did not have chance to try out the program disc which accompanies the book (for an extra £9.95), the book does state that in addition to all the program listings in the book, there are programs for a window graphics, an adventure game, a telephone bill utility and an automatic cross-referencing index utility (which was apparently used to generate the index for the book).

Overall, I liked to book very much and have been able to write some simple C programs as a result. The accompanying disc, if bought at the same time, is £2 cheaper and also offers extra programs, the book itself at £14.95 is excellent value, I only wish it was ring-bound! **A**

Printer Driver for View B3.0

John Phelan

The BASIC program VDGEN (View Driver Generator) is designed to patch VIEW B3.0 and install a default resident printer driver. (*On program disc.*)

You still have the option of loading a user-defined driver from disk to override this built-in driver, if you wish, and if you do this, you can revert to the resident driver at any time by giving the command PRINTER (without an argument).

VDGEN is set up for an Epson FX80 or compatible printer. To install the patch, you will need a ROM image of VIEW B3.0 on disk. Just type:

```
CHAIN"VDGEN"
```

to run the program. You are prompted for the source filename (your copy of VIEW B3.0) and the destination filename (the name you choose to give the patched version of the ROM which is generated by the program).

VDGEN checks for the "B3.0" text at &A842 in your ROM image and will not proceed with the installation if it doesn't find it. It also writes the necessary patch to &80C2-4 (see Archive 1.9, page 6) to disable the 2nd Processor call which is normally made by VIEW at the language entry point. In addition, the usual patch at &A8A1 to override SWR protection is inserted.

Editing the Driver

VDGEN contains a working driver for an Epson FX80, but I hope there are sufficient prompts and remarks in the program to enable you to modify the assembler code for your own printer. However, the following points will need attention:

1. Replacement code for £ character

The byte at line 2300 is the number of bytes to be sent to the printer to print a pound sign.

The FX80 in its normal configuration requires a single replacement byte, ASCII 35 (hex 23). Some printers require several bytes to switch international character sets. For example, I use my Citizen MSP-20 in IBM GRAPHICS mode. To print the £ character I need to send the following sequence (in hex):

```
1B,ASC"R",3,ASC"#",1B,ASC"R",0
```

To implement this code I would need to change lines 2290 – 2310 as follows, for example:

```
2290 .poundcode \replacement bytes for printer
2300 EQUB 07:EQUB 1B:EQUB 52:EQUB 03
      :EQUB 23:EQUB 1B:EQUB 52:EQUB 00
2310 \
```

Of course I could also write the equivalent:

```
2290 .poundcode
2300 EQUd &03521B07:EQUd &00521B23
2310 \
```

2. Changing the function definitions

Note that the "ontable" and "offtable" codes appear as matching pairs of 4-byte code sequences. If you remove or replace a sequence from one table, you should make the necessary change in the other table. Also, it is imperative that the "hcodnum" byte defined at line 2340 correctly reflects the number of functions defined (11 in the original program):

```
2330 .hcodnum \number of hilite codes defined
2340 EQUB 11
```

3. Changing the installed printer name:

Just alter the name string at line 820. I've used a * to indicate that a patched default driver is active, rather than the usual user-defined driver in ram:

```
810 .drvnam
820 EQU$ "*"FX80"
830 EQUB 13
```

Improvements

I apologise in advance for any oversight or errors in this program. I've tried to debug it thoroughly, and it seems to work. If there are any problems, I would be glad to hear from you and help sort them out.

I would also welcome ideas for further additions/improvements. The worst aspect of this driver is the limitation of only two HT markers on screen, and the constant necessity of redefining them. Perhaps someone would care to take up the challenge and show us how to extend the use of HT codes. The <shift>+<ctrl>+fn keys (10 of them) could possibly be defined to insert further HT codes in the text and be assigned functions corresponding to some of the more usual printer facilities – condensed, enlarged, elite, NLQ, etc. **A**

Two Printer Buffers

Michael Dunbar

ArcBuffer – Clares Micro Supplies – £19.95 (£18 through Archive)

Printer Module – CCD Computer Services – £12.50

The idea of a buffer is relatively simple. It is an area of memory, used for temporary storage of transmitted information that is being sent faster than it can be dealt with. Buffers are often used when sending data from a fast computer to a slow printer, but may also be used in more general communication. The buffer may be an external hardware unit or, as in the two cases examined here, it may be implemented entirely in software, using the computer's own memory and operating system facilities. In this case the inconvenience of having some memory used up for buffer space, is usually far outweighed by the speed with which the computer will be freed for other tasks. Also, software is much cheaper.

Both the Clares and CCD packages are implemented as modules. This means that they are loaded from disc into an area of memory called the RMA (relocatable module area). When activated, they claim their memory for the buffering process, but otherwise they remain 'dormant'. The user can then run other applications programs which will use the extra facilities (in this case lessening the time for which the computer is not usable while printing). As far as the application is concerned, the buffer is just another part of the operating system. However, due to the fact that some applications do such things as altering CMOS ram, and resetting the machine, it would be advisable to check that an application you are buying supports a buffer module.

First investigations reveal that ArcBuffer buffers the RS423 port (in and out), plus the parallel printer port. PModule is less versatile, it can handle only RS423 and parallel printer output.

Both implementations are loaded in with the *RMload command. ArcBuffer is automatically activated, while PModule requires the *buffon command. CCD also provide the Acorn RS423 patch for Archimedes with dickey serial ports. *RMtidy will tidy both modules, but neither will allow you to do so unless the buffer is empty. Both modules allow the size of buffers to be varied, up to the maximum available memory in the machine. PModule will give an error if an invalid size is selected, ArcBuffer will not. So it is possible to configure a 4 Mbyte buffer on a 1Mbyte machine. As pointed out in the manual, this will cause "unspecified problems"!

ArcBuffer also allows new default buffer sizes to be saved to disc, as a part of the buffer module itself. The manual does stress the dangers of saving an invalid size, (it can never be re-loaded) and the consequent importance of making a backup copy before you start.

In use, both buffers worked fine, and virtually identically. Both are able to indicate their status (bytes free and used). ArcBuffer is flushed by the standard *fx commands, while CCD has a special command, *buffpurge to do this. CCD say that this is to prevent the buffer being accidentally flushed by a spurious <escape> code from within an applications program. This might, however, make it more difficult to stop printing from within some applications.

Extra goodies

Apart from the standard function of buffering printer output, both offer extra goodies. ArcBuffer buffers any RS423 input and output, so would be handy for direct file transfer between computers (for example, Archimedes and Z88). It must be stressed though, that once information is in the buffer, it is effectively out of reach, so any error checking protocols would not work. ArcBuffer also offers a *lprint command. This comes in very handy, as does the ability to include operating system variables in *lprint strings. For example variables may be set up with names like "Italic", "Bold" etc, containing the relevant control codes.

PModule includes undistorted 2 tone graphics and text dumps. The graphics dump is up to 4 times faster than HardcopyFX, (both using a 64k buffer size). It also offers the equivalents of *type, *list, and *print but directed to the printer. These commands are more useful than one might think, but see Archive 1.9 page 7.

The manuals are both short, but after all, there is only so much that can be said about a buffer module. The CCD manual is probably slightly more comprehensive, Clares' include a section on using ArcBuffer with their own products.

In conclusion, there is little doubt that any Archimedes user who has a printer would gain by using a buffer module. Both the products reviewed here are perfectly suitable, and reasonably priced. A decision between the two would probably rest on a user's individual requirements and whether the lower cost and fast built-in dumps of PModule outweigh ArcBuffer's *lprint, and its ability to fully buffer the RS423 port. **A**

Assembly Language Programming – 4

Alan Glover

In this part we define all the conditional execution codes alluded to before. Until now we have assumed that CMP sets the flags and that no other instruction alters them. This was a simplification. The comparative instructions, of which CMP is one, do indeed always set the flags but many other instructions can set them too.

To make an instruction such as MOV R0,R1 set the flags, put an S after the instruction, e.g. MOVS R0,R1. This facility means that in some cases explicit comparisons are not needed. For example, after using MOVS you could use the EQ conditional field and its opposite NE which would indicate whether the value in R1 and R0 is zero.

The table below gives the conditional code, its meaning, what circumstances it is used in and notes about its precise derivation (some condition codes work on several flags, or have several valid combinations of flags). Often it is simpler to remember the conditional code itself which summarises the condition it isolates. When AND and OR are used in capitals they are used in their logical sense to indicate relations.

In the table, mention is made of 'signed' and 'unsigned' numbers. Representing negative numbers in binary is a problem. The usual route around it is to use the highest bit of the number as a 'sign flag' indicating that the number is positive when it is 0, and negative when it is 1. Negative numbers are stored as 0 minus the number, e.g. -1 would be &FFFFFFF.

Four ARM flags are used by conditional fields;

Z Zero flag. When set, this indicates that a result was zero or two compared numbers are identical

M The sign flag. This indicates the state of bit 31. Hence, when set, it indicates that the number is negative.

C The carry flag. This is used in two distinct ways. It can be used as an arithmetical carry, e.g. when adding 9 and 6 in decimal you carry '1' representing 10 to the next column producing 15. The arithmetical carry works in a similar way. The other usage is as part of the shift and

rotate instructions, which are covered in full detail in the next part.

V The overflow flag. This is used to indicate that a sum has caused a carry from bit 30 to bit 31, i.e. that the number has got (for example) so high that it appears to be negative number because bit 31 has been set. For example &FFFFFFF + 1 would set the V flag. The same applies for subtractions too.

Conditional Execution fields

AL Always execute. Assumed as the default by the assembler. An instruction will be executed regardless of the current flag settings.

NV NeVer execute. An instruction with this conditional field is never executed.

EQ Equal to zero. The instruction will only be executed when the previous operation (NB actually the last operation which set the flags) produced a result which was zero (or a compare when both numbers compared were identical), i.e. the instruction is executed if the Z flag is set.

NE Not Equal to zero. The instruction will only be executed when the previous operation yielded a non-zero result (or a compare of unequal numbers), i.e. the instruction is executed if the Z flag is clear.

VS oVerflow flag Set. The instruction will only be executed if a sum has caused a carry from bit 30 to bit 31, with the effect that the sign bit (if it is being used) has been altered, i.e. the instruction is executed if the V flag is set.

MI MInus. The instruction will only be executed if the sign bit (bit 31) was 1 on the previous operation. This is intended for working with signed numbers, but also general applications for bit testing (finding whether a particular bit is 1 or 0), i.e. the instruction is executed if the N flag is set.

PL PLus. As above, except that the sign bit is 0. The instruction is executed if the N flag is clear. The next group are intended for use after arithmetic operations on unsigned numbers (where bit 31 is not being used to indicate the sign of the number). They, of course, have other uses too !

HS Higher than, OR Same / **CS** Carry Set. The instruction will only be executed if the previous operation indicated that $n1 \geq n2$ ($n1$ and $n2$ represent the first and second numbers operated on (called operands)), i.e. the instruction is executed if the C flag is set.

LO Lower than / **CC** Carry Clear. The instruction will only be executed if the previous operation indicated that $n1 < n2$, i.e. the instruction is executed if the C flag is clear.

HI Higher than. The instruction will only be executed if the previous operation indicated that $n1 > n2$, i.e. the instruction is executed if the C flag is set AND the Z flag is clear.

LS Lower than, or Same. This instruction will only be executed if the previous operation indicated that $n1 \leq n2$, i.e. the instruction is executed if the C flag is clear OR the Z flag is set.

The conditions in the next group are intended for testing on the results of signed arithmetic operations, and include the overflow flag in the permutations. Since numbers can be positive or negative these conditions are each met by two or more possible flag settings. Unless you are doing something special it is easiest to remember the concept of the condition, rather than its precise construction.

GE Greater than, or Equal to. The instruction will only be executed if the previous operation indicated that $n1 \geq n2$, i.e. the instruction is executed if a) the N flag is clear AND the V flag is clear, or b) if the N flag is set AND the V flag is set.

LT Less Than. The instruction will only be executed if the previous operation indicated that $n1 < n2$, i.e. the instruction is executed if a) the N flag is set AND the V flag is clear, or b) the N flag is clear AND the V flag is set.

GT Greater Than. The instruction will only be executed if the previous operation indicated that $n1 > n2$, i.e. the instruction is executed if a) the N flag is clear AND the V flag is clear AND the Z flag is clear, or b) the N flag is set AND the V flag is set AND the Z flag is clear.

LE Less than, or Equal to. The instruction will only be executed if the previous operation indicated that $n1 \leq n2$, i.e. the instruction is executed if a) the N flag is clear AND the V flag is set, or b) the

N flag is set AND the V flag is clear, or c) the Z flag is set.

In next month's section we continue to back up last month's examples by examining three and four register instructions, and giving a full list of the possible MOV family instructions. **A**

Those who purchased Ginn's 'Assembly Language book (the 'set text' for this ARM assembly language course) through Archive should note that there is now an errata sheet available. Please send an S.A.E. if you want a copy.

First Fonts

Brian Cowan

Surprising as it may seem, I have been converted to First Word Plus for my scientific writing! The main reason is Ian Copestake's First Fonts (Scientific). There are three parts to the software package. There is a set of characters which must be downloaded to the printer before you boot up, an associated printer driver and screen definitions for all characters.

The normal Archimedes characters are redefined into a very pleasing and clear font. The scientific character set includes a number of partial characters so that symbols such as integral signs, large sigmas, all sorts of brackets, and boxes may be constructed.

The set of available characters is displayed in the window at the bottom of the screen. Those not accessible from the keyboard may be selected with the pointer.

I had one small problem when trying to get the system to run: when you download the printer character definitions the printer must be on line. This is fine, but it means you must have a sheet of paper inserted and lined up even though you don't want to do any printing. Otherwise the Archimedes locks up. Obviously you don't have to load up the printer before you start creating a text file, but it can't be done from within First Word Plus. (At least not with the present operating system.)

I have only two complaints. Firstly you have to use an LQ printer (with the current version), secondly, and this applies to all word processors with text screens, you can not "overprint" a character; that is you can't put an acc-cent on top of a character, for example. Apart from this, it is a superb package, and at £15 it is a bargain. **A**

Sprites Old and New

Adrian Look

In this article, Adrian looks at sprites as they exist on the Archimedes under Arthur 1.2 and tries to make it a little clearer how they can be used. He then goes on to discuss sprites under RISC OS 2.0 and gives a program for you to try. He has been enabled to do this because Acorn have given us a sprite extension module for use with Arthur 1.2 (which we have put on the monthly program disc) which provides many of the sprite facilities that will be available on RISC OS 2.0.

System Sprites

In this first section I shall attempt to explain how to use the system sprites and show how they can be accessed via star commands and the BASIC command PLOT &ED.

Sprite Files

System sprites are stored in an area of memory set aside by the operating system (this is known as the 'sprite area'). How much memory is determined by *Configure SpriteSize <size>. Where <size> denotes the number of 8k chunks to be used (or 32k chunks on a 440).

You can load your sprite files into this area using *SLOAD <filename>. This command will wipe all previous contents of the sprite area and load in the new file. So if you wish to keep your existing sprites and add to them from another file you must use *SMERGE <filename> as this will not clear the sprite area before loading. Saving sprites isn't difficult either; you just type *SSAVE <filename> and all the sprites in the sprite area will be saved.

There are another two commands which allow you to access sprites from a file. These load and save a sprite straight to or from the screen, i.e. they don't effect the sprite area. *SCREENLOAD <filename> will load the first sprite of a sprite file on the screen (bottom-left to be precise) and, if appropriate, it will also change the palette and mode – this is usually only done for files that have been *ScreenSaved.

When you use *ScreenSave <filename> the Archimedes will save the current graphics, along with the mode and palette at the time, as a sprite. Since a

sprite which is being *ScreenSaved or *ScreenLoaded may be any size, the Archimedes can't guarantee that the sprite width will end on a word boundary (i.e. fill up an exact number of bytes) or that the sprite will be a full screen width. As a result the Archimedes can't *LOAD or *SAVE the sprite directly onto or off the screen – this would be the quickest method. It has to use another, slower, technique. This is why these two ways of loading and saving sprites are so slow.

You could speed up the process by *SLOADing or *SMERGEing the sprite and then PLOTting it on the screen. This method is much faster, however, there has to be enough memory set aside for the sprite to be loaded (a screen could require to 160k of memory). The screen can also be saved by clearing the sprite area (using *SNEW), *SGETting a sprite from the screen and then saving the sprite area using *SSAVE.

Sprite Management Commands

*SGET <name> will get a sprite from a rectangle on the screen and put the sprite, <name>, into the sprite area. The rectangle chosen is defined by the last two graphic points visited. So if I typed:

```
MOVE 400,400
MOVE 600,600
*SGET adrian
```

A rectangle on the screen whose vertices are (400,400) (400,600) (600,400) (600,600) will be put as a sprite called 'adrian' into the sprite area.

Once you have your sprites are in the sprite area, there are several operations that you can perform on them: *SCOPY <name1> <name2>, which produces a second sprite, name2, the same as the first sprite; *SRENAME <name1> <name2>, which changes the name of the sprite from name1 to name2 and *SDELETE <name>, which deletes the sprite called name.

You can also *SFLIPX <name> and *SFLIPY <name> a sprite. This will reflect the sprite about the X-axis and Y-axis respectively.

If you want to know what sprites are presently in the sprite area then you should use *SLIST – if there are

no sprites, then the computer will return the message 'No sprites defined', otherwise the computer will generate a list of the sprites.

You can also obtain information about the sprite area itself. This is done using the *SINFO command. It will tell you how big the sprite area is, how much free space there is, and how many sprites are stored there – if the sprite area doesn't exist (no memory set aside) then the computer will return the message 'No sprite workspace'.

The only other star command to effect the sprite area is *SNEW. This will wipe the contents of the sprite area – for good! So make sure that this is really what you want to do.

Plotting Sprites

There is no star command to allow you to plot your sprites – although there is a SWI command which will do so (SYS "OS_SpriteOp",34,<name>,0,<x>,<y>,<gcol> – for those who are interested). If you wish to display a sprite you must first select it as the 'current' sprite and then plot it. This may be done using a combination of either *SCHOOSSE <name> or VDU 23,27,0,n,0,0,0,0,0 and PLOT 232-239,x,y (or the appropriate VDU codes for this plot command).

VDU 23,27,0,n will select a sprite whose name is 0-255 e.g. VDU 23,27,0,51 is the equivalent of *SCHOOSSE 5. Having chosen the current sprite you can now plot it i.e. PLOT n,x,y – different values on n giving different effects as follows:

x & y – relative

i.e. a displacement (x,y) away from the current graphics cursor position

233 Plot sprite by x,y
234 Invert sprite-sized rectangle by x,y
235 Plot sprite mask in background colour by x,y

x & y – absolute

237 Plot sprite at x,y
238 Invert sprite-sized rectangle at x,y
239 Plot sprite mask in background colour at x,y

Plot actions 233 & 237 will plot the sprite (the whole rectangle); 234 & 237 will invert a sprite sized rectangle – the sprite is not plotted; and 235 & 239 will plot the sprite 'mask'. Some sprites are

'masked', that is certain areas are considered to be transparent. In order to achieve this effect you must plot the sprite mask first and then plot the sprite with GCOL 3. e.g.

```
*SCHOOSSE sprite
GCOL 0,0
PLOT 239,500,500
GCOL 3,0
PLOT 237,500,500
```

The whole process of plotting the mask can be done with one operation by adding 8 to the GCOL value:

```
*SCHOOSSE sprite
GCOL 8,0
PLOT 237,500,500
```

I hope that this has at least given you an insight into what can be done with sprites. However, I have included an example program below, which may help clarify the article.

```
10 REM >sprprog
20
30 REM *****
40 REM * Using Sprite Commands *
50 REM * Adrian Philip Look *
60 REM * 23rd September 1988 *
70 REM *****
80
90 MODE 12:OFF
100 ON ERROR MODE 0:REPORT:END
110
120 PROCdosomedrawing
130 PROCgetasprite
140 PROCpaintthesprite
150 END
160
170 DEFPROCdosomedrawing
180 REM draw something on the screen
190 FOR repeat=0 TO 49
200 shape=RND(4)
210 CASE shape OF
220 WHEN 0 : shape=&50
230 WHEN 1 : shape=&60
240 WHEN 2 : shape=&70
250 WHEN 3 : shape=&98
260 WHEN 4 : shape=&b0
270 ENDCASE
280 rndcol=RND(7)
290 GCOL 3,rndcol
300
310 FOR point=0 TO 3
320 rndx=RND(1279)
330 rndy=RND(1024)
340 PLOT shape+5,rndx,rndy
```



```

350 NEXT point
360 NEXT repeat
370 ENDPROC
380
390 DEFPROCgetasprite
400 REM get user to input sprite
410 DIM points(1,1)
420 FOR point=0 TO 1
430   GCOL 3,15
440   PROCwaitforrelease
450   points(point,0)=x
460   points(point,1)=y
470   MOVE points(point,0),0:DRAW
         points(point,0),1024
480   MOVE 0,points(point,1):DRAW
         1279,points(point,1)
490 REPEAT
500   MOUSE x,y,b
510   MOVE points(point,0),0:DRAW
         points(point,0),1024
520   MOVE 0,points(point,1):DRAW
         1279,points(point,1)
530   points(point,0)=x
540   points(point,1)=y
550   MOVE x,0:DRAW x,1024
560   MOVE 0,y:DRAW 1279,y
570   OSCLI("FX 19")
580 UNTIL b=4
590 GCOL 0,0
600 MOVE points(point,0),0:DRAW
         points(point,0),1024
610 MOVE 0,points(point,1):DRAW
         1279,points(point,1)
620 NEXT point
630
640 REM get sprite
650 MOVE points(0,0),points(0,1)
660 MOVE points(1,0),points(1,1)
670 *SGET temp
680 ENDPROC
690
700 DEFPROCwaitforrelease
710 REPEAT
720   MOUSE x,y,b
730 UNTIL b=0
740 ENDPROC
750
760 DEFPROCpaintthesprite
770 PROCwaitforrelease
780 CLS
790 GCOL 3,0
800 *SCHOSE temp
810 MOUSE oldx,oldy,b
820 PLOT 237,oldx,oldy
830 REPEAT
840   MOUSE x,y,b

```

```

850 OSCLI("FX 19")
860 PLOT 237,oldx,oldy
870 IF b=4 THEN GCOL 0,0:PLOT 237,
         x,y:GCOL 3,0
880 OSCLI("FX 19")
890 PLOT 237,x,y
900 oldx=x:oldy=y
910 UNTIL FALSE
920 ENDPROC

```

Arthur 2.00 Sprites

Arthur 1.20 already has quite an extensive range of sprite facilities – provided via SWI OS_SpriteOp. This SWI allows us to manipulate both 'system' and 'user' sprites. System sprites are those which use the memory set aside by *Configure SpriteSize and user sprites are those which use memory provided by the program/application. Serious applications should provide memory for sprites (by a DIM statement) rather than hoping that the system sprite size will be big enough.

New life for old sprites

Arthur 2.00 promises to raise sprites to a new level. Until now designing sprites proved to be difficult because there were no graphics routines which could be used with them, e.g. a sprite couldn't be created by drawing a circle on it or by writing text to it – hence the need for the extremely cumbersome sprite editor provided on the Welcome disc. In Arthur 2.00, sprites will be able to receive VDU commands. This means that all the graphics facilities available on-screen can now be applied to sprites (without affecting the screen). Not only can this be done to the sprites, but also to their 'masks' (a sprite 'mask' denotes the areas of the sprite that are transparent, i.e. the background colour will show through).

Sprites and their masks will also be allowed to be displayed as a scaled version of the real thing – so you can enlarge or reduce your sprites to the size you require.

Other facilities that will be provided include the ability to set the pointer's shape to be a sprite, display your sprites grey scaled, and append sprites onto one another. All these facilities will be provided as a consistent extension of the existing SWI OS_SpriteOp.

We know of these facilities, not because we have an Arthur 2.00 O.S., but because we have an extension module which allows these functions to be used with Arthur 1.20. I only hope that the rest of Arthur 2.00 is as good!

Here is an example listing which uses some of the features mentioned above and shows just how easy it is to enlarge and contract sprites. (You will not be able to run this program unless you have a copy of the sprite extension module.)

```

10 REM >zoom
20
30 REM *****
40 REM * Using Sprite Extensions *
50 REM * by Adrian Philip Look *
60 REM * 23rd September 1988 *
70 REM *****
80
90 REM set up the scale block
100 DIM factor% 15
110 factor%!8=1
120 factor%!12=1
130
140 REM set up user sprite area
150 DIM spriteblock% &4000
160 spriteblock%!0=&4000
170 spriteblock%!4=0
180 spriteblock%!8=16
190 spriteblock%!12=36
200
210 MODE 12:OFF
220 ON ERROR MODE 0:REPORT:END
230
240 SYS "OS_CLI","SprExtend"
      :REM load module
250 SYS "OS_SpriteOp",&10A,
      spriteblock%,"Sprites"
      :REM and sprites
260
270 REPEAT
280   posx=RND(1279)
290   posy=RND(1023)
300   sizex=RND(10)
310   sizey=RND(10)
320
330   REM plot scaled sprite
340   factor%!0=sizex
350   factor%!4=sizey
360   SYS "OS_SpriteOp",&13A,
      spriteblock%,"palette",posx
      ,posy,8,factor%,-1
370 UNTIL FALSE

```

(Continued from opposite page)

Compatibility

I managed to get Keydefine working with First Word Plus easily enough and with Edit. However it didn't work properly with Inter-Word, though it did with Inter-Sheet. Wordwise Plus works in 40 column mode, so that was no good as keydefine assumes 80 columns and Graphic-Writer with all its hard breaks didn't give Keydefine a chance. It would be interesting to know how many other pieces of software Keydefine can be made to work with. For example, if it worked with some of the comms packages, then it would make typing in responses a lot quicker. However, Psientific Software say they haven't got much Archimedes software that they can try it out on but I'm sure they will be trying to modify it to make it compatible with other software as they are requested to do so by Keydefine users.

Conclusion

This is, potentially, an extremely useful piece of software, especially for those whose typing speed is as slow as mine. At £14.95, it is a little expensive for what it is – just a simple utility. However, even at that price, it might be good value for money for someone who writes a lot of letters with standard sentences and paragraphs or for a teacher writing reports who wants to have a lot of suggested comments in front of him/her when faced with writing 150 reports in a couple of days.

It is simple to use, simple to install, and the documentation is adequate. There is however one limitation which might be a problem for using it with some pieces of software in that it will only work on a screen size of 80 x 32 (i.e. modes 0, 8, 12 and 15.) **A**

(We agree with John that this is a useful bit of software, if a little over-priced, so we've negotiated a special discount from Psientific Software; we can offer it to Archive subscribers for £13 inclusive.)

Keydefine – A Powerful Utility

John Almann

The facility for providing pre-defined function keys on the familiar red buttons of the BBC micro-computer is something that I have appreciated over the years. Especially to a slow typist such as myself, it is very useful to be able to put commonly used text as well as control codes and other keystrokes onto the function keys. This facility is also provided, as I am sure you know, on the Archimedes computer but with its increased power over the old BBC machine, surely something more sophisticated is possible. Keydefine from Psientific Software provides the added sophistication that one might have hoped for.

The first limitation of the existing function key facility is that you can only have 12 pre-defined text sequences (13, if you count the Print key) and the amount of text is limited to about 250 characters per key. The second limitation is that you cannot see what text is programmed onto any given key and therefore you have to rely on function key strips. This is OK if you have a set of function key information which is fixed over a period of time but if you want to keep changing the information then the limitation is obvious.

The aim of Keydefine is to do away with both these limitations. This is done by holding the text and programs in a relocatable module and allowing you to access the text strings by picking them directly off the screen by using a screen bank. The software is mouse-operated and the pointer is used to point to the text required.

Explanations are given in the rather limited (but adequate) documentation as to how to use the facility with First Word Plus, ArcWriter and the BASIC Editor. The reason that separate explanations are needed is that Keydefine is activated by using the mouse buttons. You therefore have to find a mouse button combination which does not activate some other facility within the main software running. This mouse button combination, as well as the delay involved, can be set up by the use of a star command which could be added to the boot file for a given application.

Having put the cursor where the extra text is to be added and having pressed the required mouse button combination, the screen bank is switched to reveal a page of text which you have previously set up. If the text required is on this page, simply click on it and the screen reverts to the original bank with the extra text added at the cursor position.

You can use a number of pages of text, as there are page up and page down boxes at the bottom of the Keydefine screen which you can click to switch from page to page. Each time you come back to the Keydefine screen you are on the page that you used last time (and with the pointer in the same place as last time irrespective of the pointer position on the main screen). This clearly makes it a lot easier to have sets of specific texts available for different applications with the ability to switch quickly from one set to another.

Keydefine also gives you the day's date in a box at the bottom of its screen though I haven't found any way to change the format; it is presented as "Sat,10 Sep 1988".

To provide the possibility of using Keydefine with other software, you can choose to activate it from a key-press rather than from the mouse buttons. So, for instance, with ArcWriter, you initialise Keydefine with *KEYMODULE and *KEYON 220 which sets it up so that the Keydefine screen appears when you press <shift-f12>. Once the screen is available, the words can be selected with the usual mouse/pointer arrangement.

Setting up the pages of text is fairly straightforward. All you need is a WP that can save text in pure ASCII form. For example, First Word Plus will do – just save the text with WPmode OFF. However, Wordwise Plus is no good because it saves the text with carriage returns instead of linefeeds which Keydefine requires. The text, which can include control codes in the same format as the familiar programmable function keys, can be placed anywhere on the screen, provided that each piece of text is separated by one or more tilde characters ("~").

(Continued opposite)

Dragging your WIMPs

Adrian Look

This month I shall try to explain a feature of the WIMP environment known as 'dragging'. The ability to 'drag' objects around the screen is a feature of most window environments – the Arthur Window Manager (AWM) being no exception. However, the AWM can only drag boxes (not irregular shapes).

These boxes may be either 'fixed size' or 'rubber'. 'Fixed size' boxes can only have their position changed (i.e. they can be moved around the screen). Whereas 'rubber' boxes have one corner fixed and the other movable – the edges of the box behave like a rubber bands, stretching and shrinking as the box shape changes. Thus 'rubber' boxes can only have their size changed.

There is one more user drag type to introduce you to – 'invisible'. This helps the AWM overcome its inability to drag any shape other than rectangles. When the 'invisible' drag type is used the AWM does not update the screen. Instead, responsibility for drawing the dragged item is passed to the application. This way the application can drag any shape it wishes, but it must do all the processing.

The AWM has seven drag types (these also include those relevant to the operation of the AWM):

- 1 change window position drag
- 2 change size of window drag
- 3 drag horizontal scroll bar
- 4 drag vertical scroll bar
- 5 user drag – fixed size box
- 6 user drag – rubber box
- 7 user drag – invisible box

SYS 'Wimp_DragBox' at &400D0

This routine invokes a drag sequence. Basically, the routine needs the initial position of the drag box and parent box. (The parent box defines the rectangle in which the drag box is confined.) However, the AWM already knows the parent boxes for drag types 1-4. This means that for these drag types the application doesn't have to input the parent box. Thus to use the 'Wimp_DragBox' routine:

```
DIM block% &100
:
REM main program which may call
PROCinvokedrag
:
END
```

```
DEFPROCinvokedrag (type, window, x0
, y0, x1, y1, px0, py0, px1, py1)
IF type<1 OR type>7 THEN ERROR 1,
"illegal drag type"
```

```
block%!0=window
block%!4=type
block%!8=x0
block%!12=y0
block%!16=x1
block%!20=y1
```

```
IF type>4 THEN
block%!24=px0
block%!28=py0
block%!32=px1
block%!36=py1
```

```
ENDIF
SYS "Wimp_DragBox",,block%
ENDPROC
```

If you wish to cancel a drag box then you can call the 'Wimp_DragBox' routine with R1<=0. For example:

```
DEFPROCcanceldrag
SYS "Wimp_DragBox",,-1
ENDPROC
```

AWM will update the drag box (unless drag type 7 is used) until all the mouse buttons are released. When this happens, the AWM will do one of two things, depending on the drag type. For drag types 1-4, the AWM will call the 'Wimp_Open Window' routine. For drag type 5-7, the AWM will generate a 'Wimp_Poll' condition to tell the application that the drag operation has finished.

SYS 'Wimp_Poll', condition n° 7

(User Drag Finished) When this condition occurs the 'Wimp_Poll' routine will return the final co-ordinates of the drag box. The application can then do with these co-ordinates what it wishes e.g.

```
DIM block% &100
:
REPEAT
SYS "Wimp_Poll",,block% TO reason
CASE reason OF
:
WHEN 7 : PROCdragfinished
:
ENDCASE
UNTIL FALSE
END
```


Evaluating π

John Phelan

Brian Cowan writes... A few months ago I asked Archive readers about calculating π on computers (Archive 1.9). I wanted to do this to an arbitrary number of decimal places, so this had to be done (to the editor's surprise) (See what cheek I have to put up with! Ed.) using integer arithmetic. For this reason, it seemed an ideal task for the Archimedes and there were a number of responses. Only two of them were true integer calculations and so exact to the specified number of places. Terry Bromilow sent me a program that he developed on the old BBC and John Phelan sent in the following explanation and program together with a brief history of π . (Paul Denize from New Zealand sent us an article from

Scientific American explaining, amongst other things, how to calculate π to two billion places using a calculator! Ed.) For anyone wishing to take this matter further I would recommend the article by Dario Castellanos on "The Ubiquitous π ", which appeared in the Mathematics Magazine, vol.61, no.2, April 1988 starting on page 67 and continuing in the following issue.

Evaluating π

The program will evaluate π to any required number of places – time and memory permitting! The routine was originally developed on a TRS-80 (those were the days!). You may like to know that the poor old TRS-80 (which is still going strong, by the way) took over 6 hours to compute 1000 places!

```
DEFPROCdragfinished
newx0=block%!0
newy0=block%!4
newx1=block%!8
newy1=block%!12
:
REM process the result
:
ENDPROC
```

Updating an 'Invisible' Drag

The AWM does not allow the application to directly update an 'invisible' drag 'box', so in order to do this the application must keep track of what is going on itself.

The best way of doing this is to set a 'drag-in-progress flag' when the drag box is invoked and note the position of the pointer using 'Wimp_GetPointerInfo' (Archive 1.6, page 31). The application can then keep track of the pointers movements and update the drag 'box' appropriately.

```
DIM block% &100
drag=FALSE
:
REM start an invisible drag
PROCinvokedrag(7,window,x0,y0,x1,y1
                ,px1,py1,px1,py1)
REM get pointer co-ordinates
SYS "Wimp_GetPointerInfo",,block%
mx=block%!0:my=block%!4
REM set drag flag
drag=TRUE
```

```
REM the Wimp_Poll loop
REPEAT
  SYS "Wimp_Poll",,block% TO reason
  CASE reason OF
    WHEN 0 : IF drag THEN
      PROCupdatedrag
    ENDCASE
  UNTIL FALSE
END
:
DEFPROCupdatedrag
REM get pointer position
SYS "Wimp_GetPointerInfo",block%
REM quit if no change
IF block%!0=mx AND block%!4=my THEN
ENDPROC
  newmx=block%!0:newmy=block%!4
:
REM update appropriately
:
  mx=newmx:my=newmy
ENDPROC
```

Drag boxes are quite a sophisticated feature of the AWM and are only likely to be used in the more complex applications. Because of this I feel that it would be a bit wasteful filling the magazine up with a program listing. However, there will be an example program on this month's program disk, if anyone is interested. **A**

Next month: Templates – a labour saving technique.

If you're sufficiently interested you can see for yourself how the Archimedes copes. (171 seconds in RAMBASIC. Ed.)

The routine used in this program is based on a well-established formula whose terms are ARCTAN Power Series and which can be evaluated by exact integer arithmetic.

An ARCTAN (inverse tangent) power series is defined by:

$$\text{ARCTAN}(x) = 1/x - (1/x)^3/3 + (1/x)^5/5 - (1/x)^7/7 + \dots$$

The characteristics of this series are:

1. Each term involves the basic element $(1/x)$ raised to an odd power
2. Each term is divided by the power (or exponent)
3. The sign of each term alternates between + and -

It is probably well known that a reciprocal such as $1/x$ can be represented in decimal form in one of two ways:

1. An exact decimal e.g. $1/8 = 0.125$
2. A recurring decimal e.g. $1/7 = 0.14285714285\dots$

Converting a reciprocal to a decimal is done by long division e.g.

$$\begin{array}{r} 0.1428571428571\dots \\ 7 \overline{) 1.0000000000000\dots} \end{array}$$

This will probably cause a few shudders and bring back bad memories of school maths lessons! It is essentially what the routine in the program PI does. Each term is evaluated to the required number of places and sufficient terms are taken in each series so that the number of leading zeros in the last term is the same as the number of decimal places being evaluated.

The actual formula used in the program is:

$$\pi = 24 * \text{ARCTAN}(1/8) + 8 * \text{ARCTAN}(1/57) + 4 * \text{ARCTAN}(1/239)$$

This formula is coded in DATA statements in the program (lines 270-300) with three parameters for each ARCTAN term:

$24 * \text{ARCTAN}(1/8)$ is 8,24,1 (the 1 indicates a positive multiplier)

$8 * \text{ARCTAN}(1/57)$ is 57, 8,1

$4 * \text{ARCTAN}(1/239)$ is 239, 4,1

The central concept in the program is the use of arrays to simulate exact integer operations (divide, multiply, add and subtract) with multi-digit numbers. To evaluate π to 100 places, for example, we need to work with integers which are 100 digits long (in fact, a few extra digits will be needed to ensure accuracy in the 100th digit).

Multi-digit numbers are held in arrays. Each element of the array will contain a number of digits determined by a "base" factor which cuts down on the number of array elements required, thus speeding up the arithmetic. The "base" is held in variable B%, declared in line 60.

Four arrays are required - A%(), B%(), X%() and Y%(). B%() is used to build up the values of successive terms of an ARCTAN series; X%() is used to hold the accumulated sum of the series; and Y%() is used for the final accumulated total of all the series in the formula. A%() is needed as a temporary store.

To explain the method, suppose we want to evaluate $\text{ARCTAN}(1/57)$ to 20 places, using a base of 100000. Each array will require 5 elements. B%() is initialised as follows:

<u>B%(0)</u>	<u>B%(1)</u>	<u>B%(2)</u>	<u>B%(3)</u>	<u>B%(4)</u>
00001	00000	00000	00000	00000

This is the numerator 1 in $1/57$ - note that there is a notional decimal point between the first two elements. To evaluate the first term of the series we divide by 57, giving:

<u>B%(0)</u>	<u>B%(1)</u>	<u>B%(2)</u>	<u>B%(3)</u>	<u>B%(4)</u>
00000	01754	38596	49122	80701

The arithmetic is as follows:

$$1/57 = 0 \text{ carry} = 1$$

$$100000/57 = 1754 \text{ carry} = 22$$

$$2200000/57 = 38596 \text{ carry} = 28$$

$$28000000/57 = 49122 \text{ carry} = 46$$

$$46000000/57 = 80701 \text{ carry} = 43$$

This result would now be transferred to X%() and we would then proceed to divide B%() twice in

succession by 57 to produce $(1/57)^3$, followed by the operation

$A\%() := B\%() / 3$

to divide $B\%()$ by 3 and store the result temporarily in $A\%()$. From here we add $A\%()$ into $X\%()$ to complete the evaluation of the second term of the series.

As the evaluation progresses, leading elements of $B\%()$ become zero. A pointer $P2\%$ points to the first non-zero element of $B\%()$ to avoid unnecessary arithmetic and measure the required accuracy. The series is fully evaluated in our example when $P2\%$ becomes equal to 5.

Add, subtract and multiply follow the same pattern except that these will begin with the last element of the array, working from right to left in the diagram above, just as in the usual paper and pencil methods. Subtraction is done (in my program) using the trusty "add to the top and pay back to the bottom" algorithm which is now frowned on in educational circles, but has always served me very well – thank you, Mr Baker!

I hope the comments above, while probably overstating the obvious for the programmers among you, will make it easy for others to follow the program π and modify/improve it.

On the TRS-80 a base of 10000 was used; this was dictated by the nature of integer storage on this machine. BBC integers are 4 bytes long and permit a larger base to be used. However, there is still a limiting factor and the chosen value of 100000 seems to be the maximum practical one.

Note that where the X term in the ARCTAN series is sufficiently small (less than 147) the divide stage can be done in one operation e.g. with 57, we can divide by $57 \times 57 = 3249$ at each stage (after the first).

I would welcome comments from anyone who takes the trouble to analyse this routine and can suggest improvements. It would be interesting to find out how fast an ARM-code version would be.

Someone might also like to experiment with using indirection vectors instead of arrays. This would certainly cut down on array overheads, but memory is not a problem on the Archimedes and I'm not sure if indirection processing is quicker than arrays.

Finally, I have included (on the program disc) an ASCII file $\pi/10k$ which gives the 10,000 places for π produced on the Archimedes in (find out if you dare!). Just load/read it into your favourite word-processor. If you require further confirmation of accuracy, then the AEC (Atomic Energy Commission) have published a book giving 500,000 places! It makes fascinating bedside reading!

If you want to preserve your own evaluation efforts for posterity, it is a simple matter to modify π to spool the output to a disk file, or you can do this manually immediately after the program displays its output on screen. Just type "PROCout <return>" to repeat the display. To save the output to disk, you could type:

```
*SPOOL filename <return>
PROCout <return>
*SPOOL <return>
```

Here are the first 200 places, just to get you started.
 $\pi = \dots$

```
3.1415926535 8979323846 2643383279
5028841971 6939937510 5820974944
5923078164 0628620899 8628034825
3421170679 8214808651 3282306647
0938446095 5058223172 5359408128
4811174502 8410270193 8521105559
6446229489 5493038196...
```

```
10 REM >PI
20 REM INITIALISE VARIABLES
30
40 F1%=0:L1%=0:M1%=0:P2%=0:Q1%=0:
   X1%=0:K1%=0:DP%=0:count%=0
50 T=0
60 B%=100000: REM BASE
70
80 CLS:PRINT"Evaluating PI:":INPUT
   "How many places? "DP%:L%=
   DP%DIV5+2
90 DIM count% 12,A%(L%+2),B%(L%+2),
   %(L%+2),Y%(L%+2)
100 CLS:PRINT"Evaluating PI to "
   STR$DP% " places"
110 PRINT STRING$(30,"-")
120
130 TIME=0
140 READ N1%
150 FOR V%=1 TO N1%
```

```

160 PROCterm
170 NEXT
180 PRINT STRING$(30, "-")
190 T=TIME DIV100:H%=T DIV3600:M%=
    (T MOD3600)DIV60:S%=(T MOD60)
200 PROCout:REM PRINT-OUT
210 PRINT"TIME: "STR$T" ("STR$H%":
    STR$M%":STR$S%)"
220 PRINT"TERMS: "STR$!count%"/"STR$
    ! (count%+4)"/"STR$! (count%+8)
230
240 END
250
260 REM DATA FOR ARCTAN ROUTINES
270 DATA 3
280 DATA 8,24,1
290 DATA 57,8,1
300 DATA 239,4,1
310
320 REM MONITORING ROUTINE
330 DEFPROCterm
340 READ X%,M%,F1%
350 PROCarctan:!(count%+4*V%-4)=J%
    :PRINT
360 PROCmulxm
370 IF F1%=1:PROCaddyx:ELSE PROCsubyx
380 ENDPROC
390
400 REM ARCTAN (1/X)
410 DEFPROCarctan
420 PRINT RIGHT$(" "+STR$M%,3)"*
    ARCTAN(1/"STR$X%") "TAB(18)": ";
430 FOR I%=0 TO L%
440 B%(I%)=0:X%(I%)=0
450 NEXT
460 P2%=1
470 B%(1)=B%:E%=X%:PROCdivbbe
480 FOR I%=1 TO L%
490 X%(I%)=B%(I%)
500 NEXT
510
520 F%=-1:IFX%<147:E%=X%*X%:T%=0:ELSE
    E%=X%:T%=1
530 J%=1:REPEAT J%=J%+2:PRINT TAB(24,
    V%+1)RIGHT$(" "+STR$J%,6);
540 FOR N%=0 TO T%:PROCdivbbe:NEXT
550 PROCdivabj
560 IFF%=1:PROCaddxa:ELSEPROCsubxa
570 F%=F%*-1
580 UNTIL P2%>L%
590 ENDPROC
600
610 REM DIVIDE(B,B,E) - B[]:=B[]/E
620 DEFPROCdivbbe
630 C%=0
640 FOR I%=P2% TO L%
650 Q1%=B%(I%)+C%
660 Q%=INT(Q1%/E%)
670 B%(I%)=Q%
680 S%=Q%*E%
690 C%=(Q1%-S%)*B%
700 NEXT
710 IF B%(P2%)=0:P2%=P2%+1
720 ENDPROC
730
740 REM DIVIDE (A,B,J) - A[]:=B[]/J
750 DEFPROCdivabj
760 C%=0
770 FOR I%=P2% TO L%
780 Q1%=B%(I%)+C%
790 Q%=INT(Q1%/J%)
800 A%(I%)=Q%
810 C%=(Q1%-Q%*J%)*B%
820 NEXT
830 ENDPROC
840
850 REM ADD (X,A) - X[]:=X[]+A[]
860 DEFPROCaddxa
870 C%=0
880 FOR I%=L% TO P2% STEP-1
890 S%=X%(I%)+A%(I%)+C%
900 IFS%>B%:S%=S%-B%:C%=1:ELSEC%=0
910 X%(I%)=S%
920 NEXT
930 IFC%=1:X%(I%)=X%(I%)+1
940 ENDPROC
950
960 REM SUBTRACT (X,A)-X[]:=X[]-A[]
970 DEFPROCsubxa
980 C%=0
990 FOR I%=L% TO P2% STEP-1
1000 S%=A%(I%)+C%
1010 IFX%(I%)<S%:X%(I%)=X%(I%)+B%:
    C%=1:ELSEC%=0
1020 X%(I%)=X%(I%)-S%
1030 NEXT
1040 IFC%=1:X%(I%)=X%(I%)-1
1050 ENDPROC
1060
1070 REM MULTIPLY (X,M) - X[]:=X[]*M
1080 DEFPROCmulxm
1090 C%=0
1100 FOR I%=L% TO 1 STEP-1
1110 S%=X%(I%)*M%+C%

```



```

1120   IFS%>=B%:C%=INT(S%/B%):S%=S%-
                                     C%*B%:ELSE C%=0
1130   X%(I%)=S%
1140 NEXT
1150 X%(0)=C%
1160 ENDPROC
1170
1180 REM ADD (Y,X) - Y[:]=Y[]+X[]
1190 DEFPROCaddyx
1200 C%=0
1210 FOR I%=L% TO 1 STEP-1
1220   S%=Y%(I%)+X%(I%)+C%
1230   IFS%>=B%:S%=S%-B%C%=1:ELSEC%=0
1240   Y%(I%)=S%
1250 NEXT
1260 Y%(0)=Y%(0)+X%(0)+C%
1270 ENDPROC
1280
1290 REM SUBTRACT (Y,X)-Y[:]=Y[]-X[]
1300 DEFPROCsubyx
1310 C%=0
1320 FOR I%=L% TO 1 STEP-1
1330   S%=X%(I%)+C%
1340   IFY%(I%)<S%:Y%(I%)=Y%(I%)+B%:
                                     C%=1:ELSEC%=0
1350   Y%(I%)=Y%(I%)-S%
1360 NEXT
1370 Y%(0)=Y%(0)-X%(0)-C%
1380 ENDPROC
1390
1400 REM OUTPUT
1410 DEFPROCout
1420 PRINT "PI=..."STR$Y%(0)". ";
1430 Y$=""
1440 J%=0
1450 REPEAT K%=0
1460   REPEAT J%=J%+1:K%=K%+1
1470     A$=RIGHT$("00000"+STR$(Y%(J%
                                     )),5):Y$=Y$+A$
1480   UNTIL K%=10 ORJ%=L%-2
1490   PRINT TAB(2);:P%=1
1500   REPEAT PRINTMID$(Y$,P%,10)" ";
                                     :P%=P%+10:UNTIL P%=51
                                     ORP%>LENY$:PRINT
1510   Y$=""
1520 UNTIL J%=L%-2
1530 PRINT STRING$(56,"-")
1540 ENDPROC

```

A Brief History of π

The history of π is a fascinating one which has bemused mathematicians down through the ages. One of the first formulas to emerge is:

$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + \dots$$

known as Leibnitz' formula (1674). However, convergence is very slow so the formula is of little practical use.

The ARCTAN power series were developed in the 18th and 19th centuries before the advent of computers. One of the first such formulas to emerge was:

$$\pi/4 = \text{ARCTAN}(1/4) + \text{ARCTAN}(3/5)$$

which gives approximately 50-figure accuracy after about 40 terms of ARCTAN(1/4) and 108 terms of ARCTAN(3/5). Others to follow were:

$$\pi/4 = \text{ARCTAN}(1/3) + \text{ARCTAN}(1/2)$$

$$\pi/4 = 4*\text{ARCTAN}(1/5) - \text{ARCTAN}(1/239)$$

This latter formula was discovered by Machin in 1706 and used nearly two centuries later by Edward Shanks. In 1873, after working on the problem with pencil and paper on and off for twenty years or so, Shanks had produced 707 places – a record at that time. Fortunately for him, he had long passed away to his reward above when it was discovered in 1944 (on ENIAC) that he had made a mistake in the 528th digit, and all succeeding digits were wrong!

Two other well known ARCTAN series are:

$$\pi/4 = 6*\text{ARCTAN}(1/8) + 2*\text{ARCTAN}(1/57) + \text{ARCTAN}(1/239) \text{ (Stormer)}$$

$$\pi/4 = 12*\text{ARCTAN}(1/18) + 8*\text{ARCTAN}(1/57) - 5*\text{ARCTAN}(1/239) \text{ (Gauss)}.$$

The first recorded computer evaluation of π was done on ENIAC (Electronic Numerical Integrator and Computer) at the Ballistic Research Laboratory in 1949. 2037 figures were calculated in 70 hours using Machin's formula.

In November 1954 and January 1955, 3089 places were calculated on NORC (Naval Ordnance Research Calculator) at Dahlgren, Virginia.

In March 1957, a Pegasus computer at the Ferranti Computer Centre, London, calculated 10,021 places in 33 hours, but a subsequent check revealed an error and only 7480 figures were correct!

In July 1958 and IBM 704 at the Paris Data Processing Centre calculated 10,000 figures in 100 mins using a combination of Machin's formula and Gregory's Series.

In July 1961, an IBM 7090 at the London Data Centre was used to calculate 20,000 places in 39 mins only, again using Machin's formula. In the same month in New York, using a faster 7090 together with some clever programming tricks, Daniel Shanks produced 100,265 places in 8hrs 43mins (including 42 mins to convert the final result from binary to decimal). Shanks used a combination of Stormer's and Gauss' formulae above.

Subsequent recorded calculations include 250,000 places computed on an IBM 7030 at the Atomic Energy Commission in Paris, February 1966; and a year later, at the same venue, Gilloud and Dichamp, using a CDC 6600, evaluated 500,000 figures in 28 hours 10 mins and published their results in a book!

Presumably this record has been broken in more recent years. The evaluation of π is still a favourite "touchstone" for testing the speed and accuracy of new mainframes and super-computers. It's a worthy exercise for the ARC.

Have fun!


(Brian Cowan adds: The latest record is held by Yamasa Kamada of Japan. In 1987 he obtained 134 million digits, which took thirty six hours. However, he has now increased this record to 201,326,000 decimal places, which he did using a NEC SX-2 supercomputer in only six hours! He is planning to go to 400 million places next year, but so far has not found a suitable computer.)

e (the base of Natural Logs)

John adds... If you desperately need to evaluate this constant to a large number of decimal places then a possible formula is:

$$\text{EXP}(x) = 1 + x + (x^2)/2! + (x^3)/3! + (x^4)/4! + \dots$$

Putting $x=1$ gives e itself.

I vaguely remember modifying my π program many years ago to evaluate this series. Convergence is very slow, but if time and memory is no object why not have a go! For reference, the first 20 figures are: 2.71828 18284 59045 23536 0287 ... 

Enthar-7

Ian Nicholls

I have just purchased "Enthar Seven", billed as "The Mega-Adventure For The Archimedes" and marketed by Robico. The opening scenario has you sitting in the pilot's chair of a doomed interplanetary space hopper orbiting a small Earth-like planet called Enthar 7. You have to escape from the space hopper before it destroys itself, by teleporting onto Enthar 7, and then find and board an escape vessel to "head once more for the freedom of the stars".

Beaming yourself down onto Enthar 7 is not too difficult and exploring the Command Centre, which is where you re-materialise, provides one or two surprises. However, although I have found the teleport room in the Command Centre, I just cannot get the thing to work and there are seven sectors of Enthar 7 to explore yet! Enthar 7 appears, from my brief experience of it, to be well written and the descriptions provide a wealth of detail, some of which has deep significance I am sure!

It also contains a number of extremely well-executed pictures to add to the interest, but it does not seem that there are clues hidden in them since it is possible to remove them and to have a text-only game. It is also possible to turn off the colours and play the game in monochrome.

The game is not new, however, since it was released for the BBC micro in 1986. This is not a drawback in itself but the Archimedes version is not a complete re-write: Enthar 7 on the Archimedes runs under the 6502 emulator, a copy of which is loaded from the game disc as part of the start-up procedure. Again, this is not necessarily a cause for complaint in a product. The result which I personally find annoying is that all of the text remains on the disc, which is accessed every time you type a command on the keyboard: such a step was necessary for the BBC micro with only 32K of RAM, but even a model 305 Archimedes should be able to hold all of the text (about 140K) in memory. Robico should alter the program to get over this problem or provide a RAMdisc module into which the text file(s) could be copied. Otherwise, we will have to wait for Arthur version 2.0 at the end of the year, which is expected to have a full RAMdisc facility built-in.

Games Galore

Adrian Look

The Archimedes has another three games on the software scene – **Freddy's Folly** and **Orion** from Minerva Systems and **Alerion** from Dabs Press, all three at the same price of £14.95. In essence, they are all shoot-'em-up games. Orion and Alerion being the alien zapping type (with lasers, bombs and all manner of strange noises), whereas Freddy's Folly is somewhat more serene.

All three were well presented and packaged but the Minerva games (in true tradition) lacked documentation. Luckily this doesn't matter too much as the games are reasonably self-explanatory. Alerion, by contrast, was well documented and covers all the technical data and the story lines.

Orion

This was most familiar game of the three. It is another imitation of the arcade classic 'Defender'. As such, it is an enjoyable game, but if you have already played it (or its equivalents) on other machines you may find it a bit stale though they have used 'state-of-the-art' techniques to improve the realism.

The objective is to protect the astronauts who roam the landscape below your spaceship. In order to do this you are equipped with a laser cannon which kills when you score a direct hit and three 'smart bombs' which blow-up any aliens on the screen at the time. However, the aliens aren't as crude as us, when they destroy the astronauts, they don't just blast them. They carrying them off into space (listen out for the plaintive cries of "Help!" when they are

One other complaint I have is that on re-visiting a location you are given the same full description as when you first visit it. In some cases the description is written assuming this is your first visit and the repetition can become a little wearing. Even the first Acornsoft adventure for the BBC micro, "Philosopher's Quest", gave you an abbreviated description when you re-entered a location. Nevertheless, these gripes apart, Enthar 7 looks to be worth the £29.95 (or £24.95 if you hunt around) for which it is being sold. **A**

picked up!) and then merge with them to become hyper intelligent mutants which attempt to seek-and-destroy your ship. While a man is being carried off, you can rescue him by shooting the alien which is carrying him. You will also have to stop the astronaut from falling to his death by catching him and putting him safely back on the ground.

The adaptation has been well done. There are some amusing sample sounds and the graphics are far less crude than earlier versions (especially the little men at the bottom – except that they now look more like cowboys than astronauts). The graphics are very smooth and very 'Defender'-like.

In conclusion, Orion definitely has the 'Defender' feel – if you enjoy shoot-em games you'll enjoy this re-made classic. The only grumble is that it takes so long to load, as did Freddy's Folly. It took 1 minute and 15 seconds from pressing <shift-break>.

Alerion

This is another action-packed shoot-em-up game. This time your attack takes a bird's eye view (as opposed the side on view that Orion adopts). This type of game always seem more difficult to 'get the hang of', but once you do, you wonder what all the fuss was about! Alerion, at first it is very difficult to handle, but after about an hour you get more confident with the controls. Mastery however, takes a lot longer!

Judging by the self-demonstration that the computer gives, there seems to be very little going on. This is definitely a misconception! There are lots of things to think about. Not only are there the usual aliens (in all shapes, sizes, speeds, etc – 13 types altogether) but you have to keep an eye on the ground otherwise you will be shot down, run into a spike or miss your targets.

For this purpose you have two weapons: an air gun and ground bombs. The main events occur on the ground. For example, when your ship is destroyed, you will start at the last home base you flew over, so you have to look out for these. There are also F bases which, when shot 8 times, give you enough energy

to performs an FAC. FAC's are activated by pressing the space bar (which I invariably forgot to do! and as a result got blown-up quite frequently). Once you have activated them, your ship becomes immune to the on coming barrage – for a limited time, of course!

The graphics and sound are well implemented. Alerion runs in a 256-colour modes, scrolls very smoothly, and tops it all with sampled sounds. To conclude, the whole purpose of this game is to shoot everything in sight! I certainly enjoyed doing this although this may not be everybody's cup of tea.

Freddy's Folly

This game has a much slower pace than the other two and suits those of us who find the traditional computers games a bit too much. It has a good story line and excellent graphics.

Basically you are feuding with Sir Freddy Phrogg, who is very envious of the castle you have just built. So as not to be out done, Sir Freddy has bought a fleet of hot-air balloons which he hopes will be able fly over your castle and bomb it. In retaliation, you employed a gun crew to shoot the balloons down. Now, this would be too easy, so Sir Freddy also hired some battleships which try to disable your gun crew by using cannon fire.

The idea is very sound and seems the basis for an excellent game. Not only that but it has been implemented very well. The graphics are beautifully animated – there are many small pieces animation which really add polish to the game.

Unfortunately, the actual game falls short of the mark. Once you get used to it, which doesn't take long, it becomes mind-blowingly easy, even on the hardest level! (*Well, I didn't find it that easy! Ed.*) The game would be much more exciting if it kept coming up with new forms of attack (and animation).

In conclusion, the idea, graphics, and implementation are brilliant but the game becomes stale after two or three attempts. Small children may enjoy it but Dad would soon get bored! **A**

Hypercubes – A 3D Exploration

Steve Drain

A note in the August edition of Archive about A.K.Dewdney's "Computer Recreations" articles in Scientific American coincided with (a) the end of term (b) the arrival of my Archimedes and (c) the purchase of "The Armchair Universe" by the same author. I had already started to write a program to display the projections of a hypercube on my Spectrum, using Beta BASIC, and I had the necessary algorithms worked out but could only show static pictures. The transfer of this program seemed like a good project with which to get to know the Archimedes, and I hoped to be able to show some rather faster updates. The resulting speed exceeded all my expectations and the accompanying program produces real time rotations needing a huge amount of matrix calculation.

However, not everything was sweetness and light. Contrary to the 'received version' in Archive, BBC BASIC is not such a wonderful language: very flexible and adaptable, yes, and fast even on a BBC B, but not friendly or intuitive. To make good use of the machine you seem to need an intimate knowledge of the operating system calls and I found that good program structure and readability are difficult to achieve, especially when memory saving is important.

Well, Archimedes and BASIC V should have changed that, but they have the problem of upward compatibility and retain many of the awkwardnesses of the earlier machines. Nevertheless, a good editor, a much fuller set of structures and some improvements to the string handling make it worthwhile getting at all the other goodies hidden away inside Arthur.

What is a hypercube?

A hypercube is the 4-dimensional analogue of an ordinary cube. It cannot be visualised directly, but in the same way that a wire-frame cube casts a 2-dimensional shadow on a screen, a hypercube casts a 3-dimensional 'shadow', and we can see its solid shape. If we rotate a cube, the shadow changes

shape and if we rotate a hypercube the 'shadow' becomes a different solid.

A cube can be thought of as a square that has been stretched upwards and a hypercube is like a cube that has been stretched in a 4th direction that has no meaning in real life, but can be expressed mathematically. The result is that a hypercube has 16 vertices (twice as many as a cube) and 32 edges. The 4 co-ordinates of the vertices are generated from the binary representation of the numbers from 0 to 15, with '0' replaced by '-1'. When the hypercube is displayed, the edges are drawn one after the other and back to the beginning. The route to taken is called an Euler Trail and is held as a list of vertices to be visited.

A cube has 3 axes of rotation and the position it ends up in depends on the order rotations are performed (they are not commutative). For a hypercube we can define 6 axes, 3 of which are those in real space and another 3 in hyperspace. Each real rotation can be paired with a hyper rotation with which it is commutative. This makes it possible to use the mouse to produce simultaneous rotations in real and hyper space, each button selecting a different pair.

Although there can be a 3-D 'shadow' of a hypercube, there is only a 2-D screen to show it on. A simple answer is to show a shadow of the 'shadow'; the shape of the shadow depends on the projection, or perspective, used. This program uses a one-point perspective. The projection point can be moved in and out, up and down and left and right, and the projection screen can be moved in and out, using the mouse with 2 buttons pressed.

3-D screen images

However, it is far more impressive to see the true 3-D shapes and this is done by producing a pair of stereoscopic images so that each eye sees a slightly different view. There are two basic techniques for making sure each eye sees only the correct image: colour separation or spacial separation. In the first, each image is made in a different primary colour and primary filters are worn over each eye (remember old 3-D films and the red and green specs?). In this program the left eye has a red image.

In the second the images are produced side by side and some system of lenses and screens keeps them

separate. These are not essential, and most people can develop a way of bringing the 2 images together by slightly crossing the eyes and then allowing the eyes to focus beyond their normal position. The 3-D picture will suddenly snap into place, after which the brain keeps the eyes 'locked on'. All three ways of viewing the projection are provided by the program. Selection is by pressing all 3 buttons, releasing and pressing one button.

Other facilities

The other facilities provided are: to reset the hypercube to its start position after it has been thoroughly rotated; to reset both the hypercube and the projection to their start positions; to change the scale of the pictures on the screen. As it is difficult to use a keyboard when looking at these pictures, all control has been provided through the mouse and it may take a little practice to get the button presses right.

Speed considerations

Lastly there is the matter of speed. The Archimedes is working like a one-armed paperhanger getting all the calculation done to make realtime movement possible so really needs the help of RAM BASIC or the ROM speed-up procedure from issue 11, page 17. On the monthly program disc is a scrunched version of the program with all procedure calls replaced by the full code etc; this runs about twice as fast as the listing below. In the shadow mode, about 10 refreshes a second are possible, in the 3-D modes it is more like 4 a second. COS and SIN seem to be so fast that a precalculated look-up table makes little difference to the display, but simplifying the projection used did, although it removed a lot of the flexibility possible.

Have fun exploring the 4th dimension, but don't get lost!

```

10 REM >hypercube
20
30 REM HYPERCUBE by Steve Drain,
                                20th August 1988
40 REM
50 REM Displays a 3-D Projection of
                                a 4-D Hypercube.
60 REM
70 REM The hypercube may be rotated
                                in 4-D space and the projection
80 REM parameters and viewing point
                                may be changed.
```

```

90 REM
100 REM The 3-D projection may be
    viewed as a stereoscopic pair
110 REM by colour separation or by
    fused images.
120 REM
130 REM All control is by the mouse.
140 REM
150 REM The RAMBASIC Module should be
    used to enhance the display.
160
170 REM GLOBAL hypercube(), h(),
    vertex%()
180 REM rotation(), eye_spacing()
190 REM x_offset%, y_offset%,
    z_offset%, z_screen%
200 REM x_mouse%, y_mouse%, button%,
    screen%, finished%, i%,j%
210 REM display_size, x_factor,
    y_factor, projection_factor
220 REM separation, angle_step
230
240 PROCmain
250 END
260
270 DEF PROCmain
280 PROCinitialise_screens
290 PROCinitialise_variables
300 REPEAT
310   CLG
320   MOUSE x_mouse%,y_mouse%,button%
330   PROCTake_action
340   PROCdisplay_screen
350   MOUSE TO 0,0
360   PROCswap_screens
370 UNTIL finished%
380 ERROR 60,""
390 ENDPROC
400
410 DEF PROCTake_action
420 CASE button% OF
430   WHEN 1
440     PROCrotate(0,3,2,1)
450   WHEN 2
460     PROCrotate(2,0,1,3)
470   WHEN 3
480     PROCalter_projection
490   WHEN 4
500     PROCrotate(1,0,2,3)
510   WHEN 5
520     PROCset_hypercube
530   WHEN 6
540     PROCalter_viewpoint
550   WHEN 7
560     PROCset_parameters
570 ENDCASE
580 ENDPROC
590
600 DEF PROCdisplay_screen
610 CASE display_mode% OF
620   WHEN 1
630     PROCshadow_2D
640   WHEN 2
650     PROCcolour_3D
660   WHEN 3
670     PROCstereo_3D
680 ENDCASE
690 ENDPROC
700
710 DEF PROCswap_screens
720 SYS 6,19
730 SYS 6,113,1-screen%
740 screen%=NOT screen%
750 SYS 6,112,1-screen%
760 ENDPROC
770
780 DEF PROCrotate(a%,b%,c%,d%):
    LOCAL x_angle,y_angle
790 x_angle=x_mouse%*angle_step
800 y_angle=y_mouse%*angle_step
810 rotation(a%,a%)=COS(x_angle)
820 rotation(a%,b%)=SIN(x_angle)
830 rotation(b%,a%)=-SIN(x_angle)
840 rotation(b%,b%)=COS(x_angle)
850 rotation(c%,c%)=COS(y_angle)
860 rotation(c%,d%)=SIN(y_angle)
870 rotation(d%,c%)=-SIN(y_angle)
880 rotation(d%,d%)=COS(y_angle)
890 hypercube()=hypercube()
    .rotation()
900 rotation()=FALSE
910 ENDPROC
920
930 DEF PROCshadow_2D
940 h()=hypercube()
950 PROCdraw_display
960 ENDPROC
970
980 DEF PROCcolour_3D
990 GCOL 1,1
1000 h()=hypercube()
1010 PROCdraw_display
1020 GCOL 1,2
1030 h()=hypercube().eye_spacing()

```



```

1040 PROCdraw_display
1050 ENDPROC
1060
1070 DEF PROCstereo_3D
1080 ORIGIN 320,400
1090 h()=hypercube()
1100 PROCdraw_display
1110 ORIGIN 960,400
1120 h()=hypercube().eye_spacing()
1130 PROCdraw_display
1140 ENDPROC
1150
1160 DEF PROCdraw_display
1170 FOR i%=0 TO 15
1180   h(i%,0)=x_factor-(h(i%,0)-
     x_offset%)/(h(i%,2)-z_offset%)
1190   h(i%,1)=y_factor-(h(i%,1)-
     y_offset%)/(h(i%,2)-z_offset%)
1200 NEXT
1210 h()=h()*projection_factor
1220 MOVE h(0,0),h(0,1)
1230 FOR i%=0 TO 31
1240   DRAW h(vertex%(i%),0),h(
     vertex%(i%),1)
1250 NEXT
1260 ENDPROC
1270
1280 DEF PROCalter_projection
1290 z_screen%+=x_mouse%*8
1300 z_offset%+=y_mouse%*8
1310 IF z_offset%<384 THEN z_offset%
     =384
1320 PROCset_factors
1330 ENDPROC
1340
1350 DEF PROCalter_viewpoint
1360 x_offset%+=x_mouse%*8
1370 y_offset%+=y_mouse%*8
1380 PROCset_factors
1390 ENDPROC
1400
1410 DEF PROCset_parameters
1420 VDU 7
1430 button%=FNbuttons_pressed
1440 CASE button% OF
1450   WHEN 1
1460     PROCshadow_display
1470   WHEN 2
1480     PROCcolour_display
1490   WHEN 4
1500     PROCstereo_display
1510   WHEN 3
1520     PROCincrease_scale
1530   WHEN 5
1540     PROCreset_projection
1550   WHEN 6
1560     PROCdecrease_scale
1570   WHEN 7
1580     PROCfurther_options
1590 ENDCASE
1600 VDU 7
1610 ENDPROC
1620
1630 DEF FNbuttons_pressed: LOCAL
     this_button%, last_button%, dummy%
1640 REPEAT
1650   dummy%=INKEY(20)
1660   MOUSE dummy%,dummy%,
     this_button%
1670   SWAP this_button%,last_button%
1680   UNTIL this_button%=last_button%
     AND this_button%<>0
1690 =this_button%
1700 END
1710
1720 DEF PROCshadow_display
1730 IF display_mode%=3 THEN
     PROCshow_menu12
1740 display_mode%=1
1750 GCOL 5
1760 ORIGIN 512,512
1770 PROCset_factors
1780 ENDPROC
1790
1800 DEF PROCcolour_display
1810 IF display_mode%=3 THEN
     PROCshow_menu12
1820 display_mode%=2
1830 ORIGIN 512,512
1840 PROCset_factors
1850 ENDPROC
1860
1870 DEF PROCstereo_display
1880 IF display_mode%<>3 THEN
     PROCshow_menu3
1890 display_mode%=3
1900 GCOL 6
1910 PROCset_factors
1920 ENDPROC
1930
1940 DEF PROCincrease_scale
1950 display_size=display_size*6/5
1960 PROCset_factors
1970 ENDPROC

```

```

1980
1990 DEF PROCdecrease_scale
2000 display_size=display_size*5/6
2010 PROCset_factors
2020 ENDPROC
2030
2040 DEF PROCreset_projection
2050 x_offset%=0:y_offset%=0:
      z_offset%=4096: z_screen%=0:
          display_size=10
2060 PROCset_factors
2070 ENDPROC
2080
2090 DEF PROCfurther_options
2100 finished%=TRUE
2110 ENDPROC
2120
2130 DEF PROCset_factors
2140 IF display_mode%=3 THEN
2150   projection_factor=(z_screen%
      +z_offset%)*display_size/16
2160 ELSE
2170   projection_factor=(z_screen%
      +z_offset%)*display_size/10
2180 ENDIF
2190 x_factor=x_offset%/z_offset%
2200 y_factor=y_offset%/z_offset%
2210 ENDPROC
2220
2230 DEF PROCshow_menu12
2240 VDU 26: VDU 5
2250 VDU 28,33,31,39,0
2260 COLOUR 4: GCOL 4
2270 FOR i%=0 TO 1
2280   CLG: CLS
2290   RESTORE 3270
2300   FOR j%=1003 TO 32 STEP -32
2310     READ menu$
2320     MOVE 1041,j%
2330     PRINT menu$
2340   NEXT
2350   RECTANGLE 1024,0,255,1023
2360   PROCswap_screens
2370 NEXT
2380 VDU 24,0;0;1023;1023;
2390 ENDPROC
2400
2410 DEF PROCshow_menu3
2420 VDU 26: VDU 5
2430 VDU 28,0,7,39,0
2440 COLOUR 4: GCOL 4
2450 FOR i%=0 TO 1
2460   CLG:CLS
2470   RESTORE 3360
2480   FOR j%=1003 TO 811 STEP -32
2490     READ menu$
2500     MOVE 16,j%
2510     PRINT menu$
2520   NEXT
2530   RECTANGLE 0,767,1279,256
2540   PROCswap_screens
2550 NEXT
2560 VDU 24,4;4;1272;759;
2570 ENDPROC
2580
2590 DEF PROCinitialise_screens
2600 ON ERROR PROCerror
2610 OFF
2620 MODE 9
2630 COLOUR 1,240,0,0: COLOUR 2,0,240
      ,0: COLOUR 3,240,240,0
2640 COLOUR 4,160,160,160:COLOUR5,240
      240,240,144:COLOUR 6,240,240,240
2650 MOUSE STEP 1
2660 screen%=TRUE
2670 SYS 6,112,2
2680 PROCshow_menu12
2690 ORIGIN 512,512
2700 GCOL 5
2710 ENDPROC
2720
2730 DEF PROCinitialise_variables
2740 x_offset%=0: y_offset%=0:
      z_offset%=4096: z_screen%=0
2750 display_size=10: display_mode%=1
2760 PROCset_factors
2770 finished%=FALSE: angle_step=PI/
      360: separation%=6
2780 DIM hypercube(15,3), h(15,3),
      t(15,1), vertex%(31)
2790 DIM rotation(3,3), eye_spacing
      (3,3)
2800 PROCset_hypercube
2810 PROCset_vertex_list
2820 PROCset_eye_spacing(separation%
      *angle_step*2)
2830 ENDPROC
2840
2850 DEF PROCset_vertex_list
2860 RESTORE 3240
2870 FOR i%=0 TO 31
2880   READ vertex%(i%)
2890 NEXT
2900 ENDPROC
2910

```



```

2920 DEF PROCset_hypcube
2930 FOR i%=0 TO 15
2940   FOR j%=0 TO 3
2950     hypcube(i%,j%)=(i%>>j% AND
2960       NEXT                                1)*512-256
2970 NEXT
2980 x_mouse%=-separation%:y_mouse%=0
2990 PROCrotate(0,3,2,1)
3000 ENDPROC
3010
3020 DEF PROCset_eye_spacing(offset)
3030 eye_spacing(1,1)= 1
3040 eye_spacing(2,2)= 1
3050 eye_spacing(0,0)= COS(offset)
3060 eye_spacing(0,3)= SIN(offset)
3070 eye_spacing(3,0)=-SIN(offset)
3080 eye_spacing(3,3)= COS(offset)
3090 ENDPROC
3100
3110 DEF PROCerror
3120 SYS 6,112
3130 SYS 6,113
3140 VDU 26: VDU 20
3150 CLS: CLG
3160 IF ERR=60 THEN
3170   PRINT "Finished - RUN to
3180 ELSE                                     Restart"
3190   PRINT ERR,REPORT$,ERL
3200 ENDIF
3210 END
3220
3230 DATA 1,3,2,6,14,10,8,9,11,3,7,15,14,12,13,9
3240 DATA 1,5,7,6,4,12,8,0,4,5,13,15,11,10,2,0
3250
3260 DATA "BUTTONS","          SMA","          EED"
3270 DATA "      LNJ","          ","Rotate ","L/R 001"
3280 DATA "U/D 010","I/O 100","          ","Adjust "
3290 DATA "I/O 011","RST 101","X/Y 110","          "
3300 DATA "Model11","          "," THEN> ","          "
3310 DATA "Screen ","FLA 001","COL 010","STE 100"
3320 DATA "          ","Scale ","INC 011","RST 101"
3330 DATA "DEC 110","          ","Menu11"
3340
3350 DATA "BUTTONS R LUI A IRX          S FCS S IRD M"
3360 DATA "          o /// d /S/ Mode: c LOT c NSE e"
3370 DATA "          t RDO j OTY          r ALE a CTC n"
3380 DATA "          a          u          111 e          l          u"
3390 DATA "ADJUST t 100 s 110          e 100 e 110 1"
3400 DATA "MENU e 010 t 101 THEN> n 010          101 1"
3410 DATA "SELECT          001          011          010          011 1"

```

A

```

420 REM SYSTEM ERROR
430 LOCATE 9,22:PRINT"* * * * * S Y S T
440 GOSUB 370
450 END
460 REM CREATE FILE
465 LOCATE 1,10
466 INPUT "Name Directory Path to
467 IF COMMAND$="" THEN COMMAND$="\
470 FRED$="DIR "+COMMAND$+" > data
480 SHELL FRED$
490 RETURN
500 REM DRAW BOX
505 SCREEN 0,0,0
510 WIDTH 80
520 COLOR 7,0
530 CLS
540 LOCATE 4,25
550 PRINT "DIRECTORY SIZE UTILITY
560 LOCATE 5,25
570 PRINT "

```

```

580 LOCATE X,Y
590 PRINT CHR$(201);
600 FOR I%=0 TO 77
610 PRINT CHR$(205);
620 NEXT
630 PRINT CHR$(187);
650 FOR I%=2+(X-1) TO 20
660 LOCATE I%,1
670 PRINT CHR$(186);
680 LOCATE I%,80
690 PRINT CHR$(186);
700 NEXT
710 PRINT CHR$(200);
720 FOR I%=0 TO 77
730 PRINT CHR$(205);
740 NEXT
750 PRINT CHR$(188);
760 RETURN

```

Well that's all for this month folks. Don't forget - keep the letters rolling in. **A**

MS-DOS Column

Ken Biddle

This month's utility programs follow, I hope they are of some interest to you. I particularly find the 'NewPrompt' utility very useful - see what you think.

Know where you are!

Set the following up as a batch file called "NEWPROMPT.BAT"

echo off

```
prompt $e[s$e[1;1H$e[K $e[7;41m $d
$t$h$h$h$h$h$h$h $p $e [7;42m
$e[u$n$g$e[0;59;"dir/w";13p
```

The above should be typed in as two lines. The first line should be the "echo off" statement and the second should contain all of the rest typed in as only one line. (The spaces should be typed as seen)

The effect of the above will be to display at the top of the screen the Date and Time as well as informing you which directory you are currently in. One final facility it offers is that function key one has been programmed so that when you press it, the current directory will be displayed on the screen.

Directory Information

This short BASIC program is designed to analyse a designated DOS directory and display various statistics about it, i.e. (a) The size of the directory in bytes. (b) The number of files in the directory. (c) The number of sub directories.

You execute the program in the normal way and it first asks you for the full path name of the directory that you wish to analyse. It will then ask you if there are any particular file types you want analysing. To look at everything you just press <return>, otherwise enter the file type of interest (in upper case) and then press <return>.

I won't go into detail here because the program is quite simple, but you may find it interesting to look at the program and see how it works, particularly the way the directory information is read. (Using MSDOS indirection)

```
10 X=3:Y=1
20 GOSUB 500 : REM DRAW A BOX
30 GOSUB 460 : REM CREATEFILE
40 C=0: CNT=0: DIR=0
50 ON ERROR GOTO 420: REM SYSTEM ERROR
```

```
55 LOCATE 1,10:PRINT STRING$(30," ");
60 LOCATE 2,10:PRINT "(or press
   enter for all file extensions)"
70 LOCATE 1,10:INPUT "Enter
   Extension of Files you wish to
   total -> ",EXT$
80 LOCATE 6,22:PRINT"****
   Calculating Directory Size ****"
90 OPEN "i",1,"DATA.DAT"
100 INPUT #1,A$:INPUT #1,A$:INPUT
    #1,D$:INPUT #1,A$
110 LOCATE 7,22:PRINT D$
120 REM INPUT INFORMATION
130 INPUT #1,A$
140 IF EOF(1) THEN GOTO 300 : REM
    RETURN TO DOS
150 B$=MID$(A$,13,9)
160 E$=MID$(A$,10,3)
170 F$=MID$(A$,1,8):H$=MID$(A$,14,5)
180 IF F$="." THEN GOTO 120
190 IF F$=".." THEN GOTO 120
200 IF H$="<DIR>" THEN DIR=DIR+1:GOTO
    120
210 IF EXT$="" THEN GOTO 230
220 IF EXT$<> E$ THEN GOTO 120
230 REM NO MATCH FOUND
240 B=VAL(B$)
250 C=C+B: CNT= CNT+1
260 LOCATE 9,22:PRINT "The Size of
   the Directory = ";C;" bytes."
270 LOCATE 11,22:PRINT "The Number of
   Files Counted = ";CNT
280 LOCATE 13,22:PRINT "The Number of
   Sub Directories=";DIR
290 GOTO 120
300 REM RETURN TO DOS
310 LOCATE 9,22:PRINT "The Size of
   the Directory = ";C;" bytes."
320 LOCATE 11,22:PRINT "The Number of
   Files Counted = ";CNT-1
330 LOCATE 13,22:PRINT "The Number of
   Sub Directories=";DIR
340 GOSUB 370
350 SHELL"del data.dat"
360 SYSTEM
370 REM PRESS A KEY
380 LOCATE 17,22:PRINT "Press any key
   to return to DOS"
390 K$=INKEY$
400 IF K$="" THEN GOTO 370
410 RETURN
```

(Continued on previous page)

Fact-File

ACE Computing	27 Victoria Road, Cambridge, CB4 3BW. (0223-322559)
Armadillo Systems Ltd	17 Glaston Road, Uppingham, Leicester, LE15 9PX. (0572-82-2499)
Atomwide Ltd	23 The Greenway, Orpington, Kent, BR5 2AY. (0689-38852)
AVP	School Hill Centre, Chepstow, Gwent, NP6 5PH. (02912-5439)
Beard Technology	111 Evering Road, London, N16 7SL.
Beebug	Dolphin Place, Holywell Hill, St Albans, Herts, AL1 1EX. (0727-40303)
Brainsoft	22 Baker Street, London, W1M 1DF. (01-486-0321)
Cambridge Microsystems	19 Pantons Street, Cambridge, CB2 1HL. (0223-66553)
CCD Computer Services	5 Fryer's Court, Boxworth End, Swavesey, Cambridge, CB4 5RA. (0954-30279)
CJE Micros	78 Brighton Road, Worthing, W Sussex, BN11 2EN. (0903-213361)
Clares Micro Supplies	98 Middlewich Road, Rudheath, Northwich, Cheshire, CW9 7DA. (0606-48511)
Colton Software	149-151 St Neots Road, Hardwick, Cambridge, CB3 7QJ. (0954-211472)
Computer Concepts	Gaddesden Place, Hemel Hempstead, Herts, HP2 6EX. (0442-63933)
Computer Spares Ltd	61A Braemar Drive, Highcliffe, Dorset.
Contex Computing	15 Woodlands Close, Cople, Bedford, MK44 3UE. (02303-347)
Dabs Press	76 Gardner Road, Prestwich, Manchester, M25 7HU. (061-773-2413)
Dudley Micro Services	32 Osborne Road, Penn, Wolverhampton, WV4 4AY. (0902-342214)
Eeckhorn Computers BV	Poortcenter Delft, Poortweg 6, 2612 PA Delft, Netherlands. (31-15-569365)
EMR Ltd	14 Mount Close, Wickford, Essex, SS11 8HG. (0702-335747)
Fairhurst Instruments	Dean Court, Woodford Road, Wilmslow, SK9 2LT. (0625-525-694)
GEM Electronics	17 Tandragee Road, Portadown, Craigavon, BT62 3BQ.
HopeSoft	Hope Cottage, Winterbourne, Newbury, Berks, RG16 8BB. (0635-248472)
Ian Copestake Software	10 Frost Drive, Wirral, L61 4XL. (051-648-6287)
IFEL	36 Upland Drive, Plymouth, Devon, PL6 6BD. (07555-7286)
Intelligent Interfaces	43B Wood Street, Stratford-on-Avon, CV37 6JQ. (0789-415875)
Jansons	93 North Street, Stilton, Peterborough, PE7 3RR. (0733-244702)
Julian Rockey Software	Brecklands, Broad Oak, Shrewsbury, SY4 3AH.
Lingenuity	P.O.Box 10, Halesworth, Suffolk, IP19 0DX. (0986-85-476)
Minerva Systems	69 Sidwell Street, Exeter, EX4 6PH. (0392-37756)
P.R.E.S.	6 Ava House, Chobham, Surrey. (0276-72046)
Pscientific Software	37 Cottesmore Road, Hessle, North Humberside, HU13 9JQ. (0482-649187)
Robico Software	3 Fairland Close, Llantrisant, Mid Glamorgan, CF7 8QH. (0443-227354)
Silicon Vision Ltd	Signal House, Lyon Road, Harrow, Middlesex, HA1 2AG. (01-422-2274)
Software Services	65 South Mossley Hill Road, Allerton, Liverpool, L19 9BG. (051-427-7894)
Spacetech	21 West Wools, Portland, Dorset, DT5 2EA. (0305-822753)
Texellence	2 Greenhill Road, Coleraine, N Ireland, BT51 3JE.

Norwich Computer Services 18 Mile End Road, Norwich, NR4 7QY. (0603-507057)

Archive

Subscription Magazine and Support Group for *Archimedes* Users

Archive Magazine contains:

- News
- Reviews
- Hints and Tips – a major feature
- Articles for Beginners
- The Latest Technical Information
- Program Listings
- Free Small Ad's Section
- HELP – Requested and Offered
- Contact Box – to help you form common interest groups

Eureka! – Bulletin Board

0603-250689 on 1200/75 or 300/300

- Very Latest News
- Down-load Software
- Mailbox Facilities
- Chat line

Technical Help Service (£8 / year)

A telephone hot-line service for immediate help with your technical problems. Any member can send written enquiries, but for a fast response use the THS!

Members' Discount: 7.5% off software from Computer Concepts, Minerva, Clares, Acorn and others when purchased through Norwich Computer Services.

Subscription: 12 issues £14.50 (UK)
Europe £20, Middle East £25,
America / Africa £28, Elsewhere £30.
Technical Help Service £8

N.B. All earlier issues have now been re-printed – you may back-date your subscription as far as issue 1 (October 1987) – to take advantage of this huge bank of information.

Archimedes is a trademark of Acorn Computers Ltd.

* Please send copies of *Archive* magazine for one year starting from
Volume ____ Issue ____

* Please enrol me on the Technical Help Service for one year. (£8)

I enclose a cheque for £ _____ payable to "Norwich Computer Services".

Name: _____

Address: _____

Postcode: _____

Norwich Computer Services, 18 Mile End Road, Norwich, NR4 7QY

Special two-year subscription starting from volume 1, issue 1 – £27